ANKARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

BİLİŞSEL RADYO ŞEBEKELERİNDE YAPAY SİNİR AĞLARI İLE SPEKTRUM YÖNETİMİ

Erdem AĞAOĞLU

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2010

Her hakkı saklıdır

Erdem AĞAOĞLU tarafından hazırlanan "**Bilişsel Radyo Şebekelerinde Yapay Sinir Ağları İle Spektrum Yönetimi**" adlı tez çalışması 02.02.2010 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman **:** Yrd. Doç. Dr. Murat Hüsnü SAZLI

Jüri Üyeleri **:**

Başkan: Yrd. Doç. Dr. Süleyman TOSUN
Ankara Üniversitesi Bilgisayar Mühendisliği ABD

Üye **:** Yrd. Doç. Dr. Hakkı Alparslan ILGIN
Ankara Üniversitesi Elektronik Mühendisliği ABD

Üye **:** Yrd. Doç. Dr. Murat Hüsnü SAZLI
Ankara Üniversitesi Elektronik Mühendisliği ABD

Yukarıdaki sonucu onaylarım

Prof. Dr. Orhan ATAKOL
Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

BİLİŞSEL RADYO ŞEBEKELERİNDE YAPAY SİNİR AĞLARI İLE SPEKTRUM YÖNETİMİ

Erdem AĞAOĞLU

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Murat H. SAZLI

Günümüzde sınırlı bir kaynak olan elektromanyetik spektrumun kullanımı genellikle devletler tarafından belirlenen kurumlar vasıtasıyla lisanslama yoluyla gerçekleştirilmektedir. Ancak bu kullanım, söz konusu spektrumunun lisanslanan kişi ya da kurum tarafından bütün zamanlarda kullanılmadığı gerçeğinden hareketle etkin bir yöntem olarak görülmemektedir.

Bilişsel radyo sistemleri, lisanslı kullanıcıların spektrumu kullanmadığı zamanlarda bu kaynağı başka kullanıcılara sunabilen bir yapı önermektedir. Yazılım tanımlı radyolar vasıtasıyla işlem gören bu sistemler, var olan spektrum kullanımını algılayacak, kullanıma uygun alanları lisanssız kullanıcılara verebilecek ve hatta muhtemel değişikliklerde kullanıcıların kesintisiz iletişimini sağlayacak alt-sistemleri içermelidir.

Bu çalışma, spektrum yönetimi olarak adlandırılan ve kullanılabilir kanalların lisanssız kullanıcılara tahsis edilmesi amacını güden yapının, destekleyici öğrenme ve bu kavram dâhilinde yer alan bir yapay sinir ağı vasıtasıyla gerçekleştirilmesini araştırmıştır.

Şubat 2010, 59 Sayfa

Anahtar Kelimeler: Bilişsel radyo, Spektrum yönetimi, Yapay sinir ağları, Destekleyici öğrenme

ABSTRACT

Master Thesis

SPECTRUM MANAGEMENT IN COGNITIVE RADIO NETWORKS USING NEURAL NETWORKS

Erdem AĞAOĞLU

Ankara University
Graduate School of Natural and Applied Sciences
Department of Electronics Engineering

Supervisor: Asst. Prof. Dr. Murat H. SAZLI

Electromagnetic spectrum, a limited resource, is usually utilized by licensing a part of the spectrum to the parties or organizations by government assigned authoritative agencies. This process seems to be an ineffective method for utilization due to the fact that these licensee parties or organizations do not actually use this licensed part of spectrum at all times.

Cognitive radio systems propose a structure for assigning this resource to the other users when it is not being used by the licensed users. The system, which operates on software defined radios, should have sub-systems to detect spectrum utilization, assign the available parts to the unlicensed users and maintain uninterrupted communication between users in a probable situation of configuration change.

This research studies the realization of the structure named spectrum management which seeks to assign available parts of the spectrum to the unlicensed users with a neural network that approximates the Q value in the reinforcement learning studies.

2009, 59 pages

Key Words: Cognitive radio, Spectrum management, Neural networks, Reinforcement Learning

TEŞEKKÜR

Tez konumun belirlenmesinin ilk gününden bu yana çalışmalarımın her safhasında yakın ilgi ve önerileri ile beni yönlendiren, iş hayatımdaki yoğunluk sebebiyle çalışmalarımda oluşan aksamalarda büyük sabır gösteren, tez danışmanım Sayın Yrd. Doç. Dr. Murat H. SAZLI'ya (Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı) teşekkürlerimi sunarım.

Tez çalışmalarım sırasında iş hayatımdaki aksaklıklara göstermiş olduğu sabırdan dolayı müdürüm Sayın Emrah ÖZÇELEBİ'ye, aksamalarda benden destek ve yardımlarını esirgemeyen mesai arkadaşım Burak DALGIÇ'a teşekkür ederim.

Son olarak yakın arkadaşlarım Koray KARAKUŞ ve Derya ÇEVİK'e bana sağladıkları moral ve destek için teşekkür ederim.

Erdem AĞAOĞLU
ANKARA, Şubat 2010

İÇİNDEKİLER

ÖZET ..... **i**

ABSTRACT ..... **ii**

TEŞEKKÜR ..... **İİİ**

KISALTMALAR DİZİNİ ..... **Vİ**

ŞEKİLLER DİZİNİ ..... **Vİİ**

ÇİZELGELER DİZİNİ ..... **Vİİİ**

1. GİRİŞ **1**

2. KURAMSAL TEMELLER **2**

2.1 Bilişsel Radyo **2**

2.1.1 Bilişsel kavram **2**

2.1.2 Tanım **3**

2.1.3 Genel bakış **3**

2.1.4 Spektrum algılama **5**

2.1.5 Bilişsellik **10**

2.1.6 Yazılım Tanımlı Radyo **12**

2.2 Yapay Sinir Ağları **13**

2.3 Destekleyici Öğrenme **18**

2.4 Q-öğrenme **23**

3. MATERYAL VE YÖNTEM **26**

3.1 Materyal **26**

3.2 Yöntem **27**

3.2.1 BR sistemi başlangıç durumları **27**

3.2.2 Yapay sinir ağının başlatılması **28**

3.2.3 Sistemi tetikleyici faktörler 28

3.2.4 Geçilebilecek durumların hesaplanması ve seçilmesi 29

3.2.5 En küçük Q değerinin hesaplanması..... 34

3.2.6 İkincil kullanıcıların yerleştirilmesi 35

3.2.7 Öğrenme..... 39

4. ARAŞTIRMA BULGULARI VE TARTIŞMA 40

4.1 Yapay Sinir Ağının Öğrenmesi..... 43

4.2 Birincil Kullanıcı Değişiklikleri 45

4.3 İkincil Kullanıcı Gereksinim Değişiklikleri..... 47

4.4 İkincil Kullanıcı Eklenmesi ve Çıkarılması 48

5. SONUÇ 50

KAYNAKLAR 51

EKLER..... 54

EK 1 YAPAY SİNİR AĞININ ÖĞRENMESİ BENZETİM SONUÇLARI 55

EK 2 BİRİNCİL KULLANICI DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI . 56

EK 3 İKİNCİL KULLANICI GEREKSİNİM DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI 57

EK 4 İKİNCİL KULLANICI EKLENMESİ YA DA ÇIKARILMASI BENZETİM SONUÇLARI 58

ÖZGEÇMİŞ..... 59

KISALTMALAR DİZİNİ

BR	Bilişsel Radyo
CBR	Olaylara Dayalı Akıl Yürütme (Case-Based Reasoning)
CPE	Öncül Tüketici Cihazı (Consumer Premise Equipment)
DARPA	Savunma Bakanlığı İleri Araştırma Projeleri Ajansı (Defense Advanced Research Projects Agency)
DSA	Dinamik Spektrum Atama
FCC	Federal İletişim Komisyonu (Federal Communications Commission)
MDP	Markov karar süreci (Markovian Decision Process)
MLP	Çok-Katmanlı Perceptron (Multi-Layer Perceptron)
QoS	Servis Kalitesi (Quality of Service)
RBFN	Radyal Taban Fonksiyon Ağı (Radial Basis Function Network)
RF	Radyo Frekansı
SNR	Sinyal-Gürültü Oranı (Signal-to-Noise Ratio)
WRAN	Kablosuz Bölgesel Alan Ağları (Wireless Regional Area Network)
XG	Yeni Nesil (neXt Generation)
YR	Yazılımsal Radyo
YSA	Yapay Sinir Ağı
YTR	Yazılım Tanımlı Radyo

ŞEKİLLER DİZİNİ

Şekil 2.1 Dinamik Spektrum Erişimi 4
Şekil 2.2 Spektrum Boşlukları 7
Şekil 2.3 Uyumlu Süzgeç Algılaması 8
Şekil 2.4 Enerji Algılaması 8
Şekil 2.5 Öznitelik Algılaması 9
Şekil 2.6 Basitleştirilmiş bilişsellik döngüsü 11
Şekil 2.7 McCulloch–Pitts nöron modeli 15
Şekil 2.8 Aktivasyon Fonksiyonları 16
Şekil 2.9 Öğreticili Öğrenme 17
Şekil 2.10 Öğreticisiz Öğrenme 17
Şekil 2.11 Destekleyici Öğrenme 19
Şekil 2.12 Yaklaşık kural döngüsü 23
Şekil 2.13 Hedef Q değerine YSA ile yaklaşma 24
Şekil 3.1 Tüm yapı durum-geçiş diyagramı 28
Şekil 3.2 İşleyen durum yapısı 30
Şekil 3.3 Geçiş durumları hesaplama algoritması 33
Şekil 3.4 Her girdinin bir kullanıcıyı gösterdiği YSA yapısı 34
Şekil 3.5 Her girdinin bir kanalı gösterdiği YSA yapısı 35
Şekil 4.1 Benzetim başlangıcı 41
Şekil 4.2 İlk hesaplamalar 42
Şekil 4.3 Gerçekleştirilen İşlemler 43
Şekil 4.4 Bekleyen durum 43
Şekil 4.5 Birincil Kullanıcı Değişikliği Deney 2 46

ÇİZELGELER DİZİNİ

Çizelge 3.1 Tetikleyici saptama tablosu..... 30
Çizelge 4.1 Yapay sinir ağı öğrenmesi benzetimi performansı 44

1. GİRİŞ

Radyo bağlantıları ile çalışan kablosuz iletişim sistemleri, sınırlı bir kaynak olan elektromanyetik spektrumun bir kısmını belli bir süre için kullanarak çalışmaktadırlar. Bu sınırlı kaynağın kullanımı günümüzde genellikle devletlere bağlı kurumlar tarafından düzenlenmektedir. Bu kurumlar spektrumu, şebeke işletmecileri ya da farklı tip organizasyonlara lisanslamaktadırlar. Ancak kaynağın bu şekildeki kullanımı iletişim ihtiyacının giderek arttığı günümüzde spektrum kıtlığına sebep olmaktadır. Bu problemi aşmak için öne sürülen Bilişsel Radyo (BR) sistemi kaynağın gerekli olduğu yerde gerekli olduğu süre boyunca kullanılması esasına dayanır.

BR, bir Yazılım Tanımlı Radyo (YTR) sisteminin zeki bir şekilde işletilmesiyle mümkün olsa da sadece YTR'ye bağlı olduğunu söylemek çok doğru değildir. Etkin bir şekilde çalışacak BR sistemi veri alışverişi için YTR kullanıyor olsa da diğer işlemler için farklı sistemlere ihtiyaç duyacaktır. Bu sistemler kullanılabilir spektrumun algılanması, sezilmiş spektrumun yönetimi ve çalışma parametrelerindeki değişikliklerinin uygulanabilmesi gibi işlemleri kapsayacaktır.

Bu tezde kullanılabilir spektrumun sezildikten ve spektrum boşlukları saptandıktan sonra yapılması gereken spektrum yönetimi işleminin Yapay Sinir Ağıları'yla (YSA) gerçekleştirilebilecek bir uygulaması geliştirilmiştir. Geliştirilen sistem, makine öğrenmesi alanının bir alt konusu olan destekleyici öğrenme (reinforcement learning) yöntemlerinden Q-öğrenme yöntemini baz almıştır. YSA bu sistem dâhilinde Q parametrelerinin tahmini için kullanılmıştır.

BR sistemi şüphesiz ki spektrum kıtlığını aşabilmek adına önerilen önemli bir yapıdır ve gerçeklenebilmesi birçok sistemin düzenli bir şekilde işlemesine bağlıdır. Bu tez, destekleyici öğrenme kavramlarını dinamik spektrum yönetimi problemine uygulama biçimi açısından tektir ve bazı pratik uygulamalar için uygulamanın şartlarına bağlı olarak gerçeklenebilir bir yapı önermektedir.

2. KURAMSAL TEMELLER

2.1 Bilişsel Radyo

2.1.1 Bilişsel kavram

İletişim problemlerini çözebileceği düşünülen bilişsel iletişim sistemleri, temellerini psikolojiye borçludurlar. İnsanların düşünme yapısını açıklamaya çalışan bilişsellik, güçlü ve etkili bir model olduğundan birçok alanda ilgi uyandırmış ve uygulama alanı bulmuştur.

Bilişsel psikoloji, insanın tüm zihinsel faaliyetlerinin altında bilişsel sürecin olduğu fikrine dayanır. Örneğin öğrenme işlemi öncelikle bilginin algılanması sonra anlaşılması ve sonra bir bilgi ile ilişkilendirilmesi şeklinde modellenebilirken, karar alma işlemi algılanan bir durumun var olan bilgilerle yorumlanıp uygulanması olarak düşünülebilir.

Davranışsal ekonomi ve davranışsal finans, sosyal yapıların hem duygusal hem de bilişsel faaliyetleriyle, ekonomik kararları ve bu kararların piyasalara olan etkilerini ilişkilendirmeye çalışır. Pazar hareketlerinin insanların rasyonel ya da irrasyonel davranışlarına göre yorumlayan alanlar, analizler esnasında bilişsel süreci sıkça kullanmaktadırlar.

Bilişsel dilbilimi, insanların dil oluşturma, öğrenme ve kullanma yapılarının en iyi bilişsellik ile açıklanabileceği temeline kurulmuş bir dilbilimi alt dalıdır. Bu dal zihinde otonom bir dil işleme merkezi olduğu fikrini reddeder. Zihnin dilbilgisini sebeplendirme yoluyla anladığını ve dil öğreniminin dil kullanımıyla gerçekleşebileceğini savunur.

Bilgisayar bilimleri ve enformasyon teorisi yapay zekâ, işbirlikçi zekâ ve robotik gibi alanlarda yaşayan varlıkların bilişsel kapasitelerini sağlamaya çalışır. Bir yerde yapay bir sistem tarafından edinilen bir deneyimin başka bir yerde başka bir yapay sistem

tarafından kullanılabilmesi problemi, çözüldüğünde şüphesiz ki yeni olanaklar getirecektir.

Ayrıca beyin faaliyetlerini inceleyen nöroloji, kişi etkileşimlerini inceleyen ergonomi ve özellikle kuantum fiziğinde anlam bulan ama bilimde genel bir kavram olarak düşünülen izleyici etkisi gibi birçok konu, bilişsel sürecin sıklıkla uygulanmaya çalışıldığı alanlardandır.

2.1.2 Tanım

İlk olarak 1998 yılında Joseph Mitola tarafından kullanılmış bir terim olan BR, yine Mitola'nın tanımı ile YTR üzerine herhangi bir hesaplama zekâsının bütünleştirilmesidir (Mitola 2006). Konu hakkında yayınlanan ilk makalede ise, "radyo frekans (RF) bantları, hava arabirimleri ve protokoller gibi parametreler hakkında, radyo modeli tabanlı akıl yürütebilen Yazılımsal Radyo (YR) sistemi" olarak tanımlanmıştır (Mitola vd. 1999). Ayrıca BR, "YTR üzerinde çalışan, bulunduğu ortamdan haberdar olan, öğrenmek ve girdisindeki istatistiksel varyasyonlara uyum sağlamak için 'yaparak anlama' yöntembilimini kullanan, akıllı kablosuz iletişim sistemi" gibi birçok benzer tanıma sahiptir (Haykin 2005).

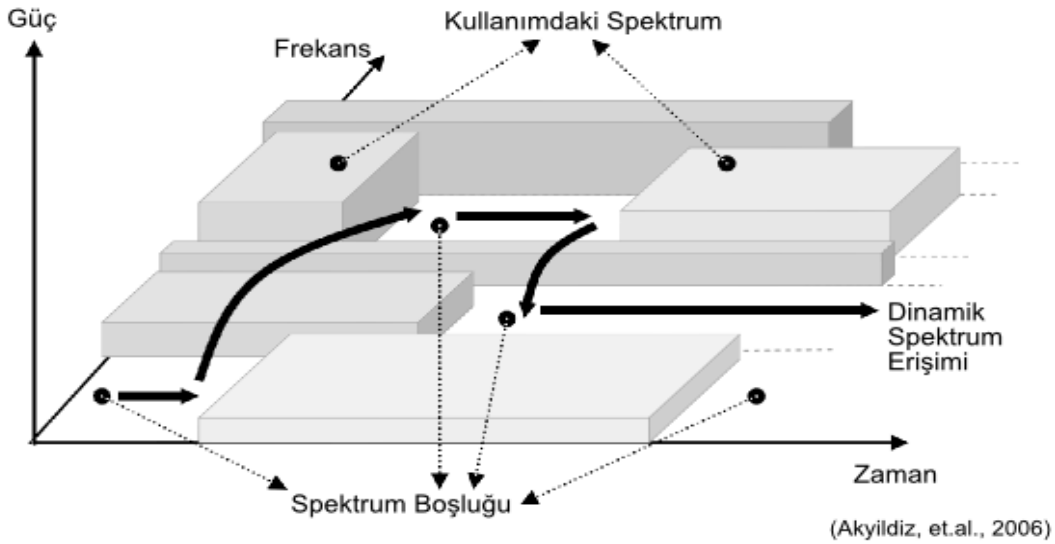
2.1.3 Genel bakış

Tanımlar farklı da olsa genel olarak BR sistemlerinin hedeflediği amaçlar; iletişim problemlerinin daha etkili bir şekilde çözülmesi, gerektiği an gerektiği yerde daha güvenilir iletişim ve başta spektrum olmak üzere radyo kaynaklarının daha etkin kullanımıdır. Bu amaçlara ulaşabilmek adına bilişsel süreci kullanan BR yapısının, yapması gerekenler şu şekilde özetlenebilir (Kushwaha vd. 2008):

- Ortam koşullarının izlenmesi / RF girdisinin algılanması
- Parametrelerinin değerlerinin hesaplanabilmesi / Bilişsellik – Yönetim
- Radyo çalışma parametrelerinin değiştirilebilmesi / Uygulama

Bu konular bu halleriyle ifade edildiğinde gayet soyutturlar ve her soyut konu gibi birçok yönden ele alınabilmektedirler. Eğer inceleme herhangi bir iletişim problemi (örn. sinyal-gürültü oranı (SNR), bağlanırlık) açısından gerçekleştiriliyorsa bu başlıklar, ortam koşullarının ilgili iletişim problemi bağlamında izlenmesi, çalışma parametrelerinin bu problemi aşacak şekilde düzenlenmesi ve bu parametre değerlerinin uygulanabilmesi haline gelebilmektedir. Ancak inceleme kaynakların kullanımının (örn. spektrum etkinliği, enerji gereksinimi) optimize edilmesiyle ilgili ise o zaman başlıklar, ortam koşullarına bağlı kaynakların izlenmesi, kullanılan kaynaklar ile ilgili değerlerin hesaplanması ve altyapının kullanımının düzenlenmesi şeklinde ele alınmalıdır. Dolayısıyla geliştirilebilecek konular ilgi çekici derecede fazlasıyla çeşitlidir.

Yukarıdaki tanımlar her ne kadar geniş kapsamlı da olsa çalışmalar özellikle DARPA (ABD Savunma Bakanlığı İleri Araştırma Projeleri Ajansı - Defense Advanced Research Projects Agency) XG (Yeni Nesil – neXt Generation) programı ve FCC'nin (Federal İletişim Komisyonu – Federal Communications Commission) konuya ilgisiyle beraber spektrum kullanımının optimize edilmesi konusuna yoğunlaşmıştır (Le vd. 2007). Dinamik spektrum atama (DSA) (Şekil 2.1) problemi için çözüm teşkil edebilen BR sisteminin yapması gerekenler bu yapı içinde; spektrum algılayıcı, spektrum yönetici ve spektrum uyumlu radyo şeklinde incelenmiştir (Akyildiz vd. 2006).



Şekil 2.1 Dinamik Spektrum Erişimi

2.1.4 Spektrum algılama

Spektrum algılama işlemi BR sistemi dahilinde spektrum bilinirliği yaratmak amacına hizmet ettiğinden, teorik olarak uygulanabilir tek işlem değildir. Spektrum bilinirliği normalde var olan sistemlerin zaten uygulanmış olan bazı işlemlerinin kullanılmasıyla elde edilebilir. Sistem içindeki her bir eleman kendi spektrum kullanımını diğerlerine raporlayabilir. Spektrum kullanımının sadece vericiler tarafından değil alıcılar tarafından da raporlanması paylaşım işlemini bir adım daha öteye götürecektir. Teorik olarak raporların bu şekilde toplanması işlemi şebekenin tüm elemanlarının herhangi bir iletim yapmaya çalıştıklarında karşılaşacakları ya da oluşturacakları girişimi bilmelerini sağlar. Fakat bu işlem, spektrum kullanımın homojen olduğu ve tek bir varlık tarafından kontrol edildiği uygulamalar için geçerlidir (Marshall 2006).

Dolayısıyla BR sisteminin çeşitli altyapılar üzerinde çalışabilmesi, spektrum bilinirliğini algılama yöntemiyle sağlamasına bağlıdır. Spektrum algılama, bir referans bandın frekans kullanım bilgisi ve bir frekansta kullanılan hava ara yüzü bilgisinin toplanıp nicel ve nitel analiz edilmesi işlemi olarak tanımlanabilir (Gandetto vd. 2007). Algılama işlemi fazlasıyla ilgilenilen probleme bağlıdır ve problemlerin geneli ile ilgili söylenebilecek herhangi bir fikir, bazı konularda anlamlı olabilirken diğerlerinde anlamlı olmayabilir. Bu doğrultuda RF girdi algılaması aşağıdakileri kapsayabilmektedir (Kushwaha vd. 2008):

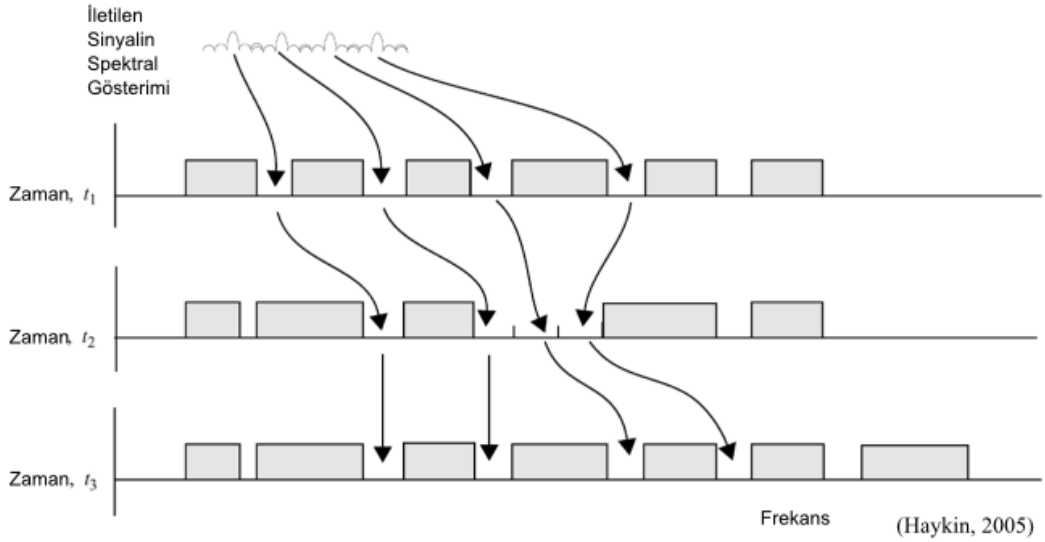
- Radyo ortamının girişim sıcaklığının kestirilmesi
- Kanal durum bilgisinin kestirilmesi
- Verici tarafından kullanılabilecek kanal kapasitesinin tahmini
- Spektrum boşluklarının sezilmesi

Girişim sıcaklığı, FCC Spektrum Kuralları Görev Gücü (Spectrum Policy Task Force) tarafından önerilmiş sadece verici tabanlı sabit girişim engelleme işlemleri yerine, gerçek zamanlı alıcı – verici etkileşimlerine dayanan uyarlamalı operasyonların metriğidir (Haykin 2005). Girişim sıcaklığının bir radyo ortamındaki girişim kaynaklarını niceleyip yönetebilmesi amaçlanmıştır. Ayrıca girişim sıcaklığı sınırı

belirtimleri belli bir coğrafi alanda belli bir frekans bandında alıcının tatminkâr bir şekilde çalışabileceği değerleri tanımlar. Şüphesiz ki çalışılacak frekans bandında elde edilecek performans bu değerle doğrudan ilişkilidir ve kestirimi sayesinde gerekli performans düzenlemesi yapılabilir. Gerekli işlem, iç gürültü kaynaklarının ve dış RF enerji kaynaklarının toplamsal dağılımından bir multitaper metotla girişim sıcaklığının güç spektrumunun kestirimi ya da, uygun olduğu yerlerde birden fazla algılayıcı kullanılmasıyla gerçekleştirilebilir. Haykin (2005), çalışmasında multitaper spektral kestirim ile tekil değer ayrıştırması (Singular Value Decomposition, SVD) yönteminin birleşimi ile bir RF ortamındaki gürültü tabanının güç spektrumunun etkin bir şekilde kestirilebildiğini göstermiştir. Girişim sıcaklığı kestirimi, birincil kullanıcıların katlanabileceği girişim miktarının saptanabilmesi açısından önem teşkil etmektedir. İkincil kullanıcılar saptanan değeri aşmadıkları sürece o bantta iletim gerçekleştirebilirler (Akyildiz vd. 2009).

Kanal durum bilgisi her iletişim bağlantısında olduğu gibi BR bağlantılarında da kanal kapasitesinin hesaplanması için gerekli bir bilgidir (Haykin 2005). Bu işlem için daha önceleri uygulanan diferansiyel sezim ve pilot sinyal iletimi yöntemleri kullanılabilir. Ancak bu yöntemlerin bazı eksiklikleri mevcuttur ve bu eksiklikler bazı durumlarda istenmeyen durumlar meydana getirebilir. Muhtemel problemlerden kaçınmak adına, alıcının bir öğreticili öğrenme ile kanal kestirimi yaptığı bir de yaptığı kestirimle veri iletimi gerçekleştirdiği izleme aşamaları arasında değişerek çalışması yani yarı-kör eğitim yöntemi öne sürülmüştür.

Diğer algılanabilecek tüm parametreler arasında en önemli olanı ikincil kullanıcılar tarafından direkt olarak kullanılacak bantların yani spektrum boşluklarının tespitidir. Spektrum boşluğu, birincil ya da lisanslı kullanıcıya atanmış ancak belli bir zamanda ve coğrafi bölgede bu kullanıcı tarafından kullanılmayan frekans bandıdır (Haykin 2005). Bu boşluğun ikincil ya da lisanslı olmayan kullanıcının kullanımına sunulmasıyla spektrum ciddi oranda daha etkin kullanılabilir (Şekil 2.2). Radyo izleme ve elektronik cihazlardan bilindiği üzere, boş kaynakların tespiti için en iyi yol zaman/frekans düzleminin incelenmesidir (Jondral 2007). Dolayısıyla spektrum boşluklarının bulunması işlemi, kavramsal olarak, geniş bir spektrumu tarayan alıcılarla ya da tüm



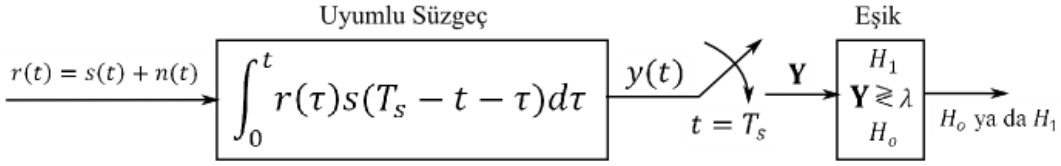
Şekil 2.2 Spektrum Boşlukları

aralığı kapsayacak bir süzgeç bankasıyla gerçekleştirilebilir. Bu iki yöntemin de avantajları ve dezavantajları bulunmaktadır. Süzgeç bankaları karmaşık ve pahalı aygıtlardır. Tarama işlemi ise geniş spektrum alanları için uzun sürebilmektedir. Fakat BR'nin aktif olmadığı zamanlarda tarama yaptığı, bu bilgiyi kaydettiği ve bir bağlantı kurulmaya çalışıldığında bu bilginin kullanıldığı çalışma yapısında bu süre önemini kaybetmektedir.

Diğer taraftan birincil kullanıcıların algılanması olarak değerlendirilebilecek spektrum boşluklarının tespiti işlemi, çalışmalarda, genellikle vericilerin algılanması şeklinde araştırılmıştır. Uyumlu süzgeç algılaması (matched filter detection), enerji algılaması (energy detection) ve dönemli-durağan öznitelik algılaması (cyclostationary feature detection) öne çıkan yöntemler arasındadır.

2.1.4.1. Uyumlu Süzgeç Algılaması

Uyumlu süzgeç, beyaz Gaussian gürültü varlığında alınan işaretin SNR değerini maksimize eden bir doğrusal optimal süzgeçtir. Uyumlu süzgeç, bilinen bir işaretin ya da taslağın bilinmeyen bir işaret içinde varlığını tespit edebilir (Şekil 2.3). Radar uygulamalarında sıklıkla yer alan uyumlu süzgeçlerin BR sistemlerinde uygulanma imkânı, birincil kullanıcıların işaretinin bilinmesinin çok kolay olmaması sebebiyle



Şekil 2.3 Uyumlu Süzgeç Algılaması

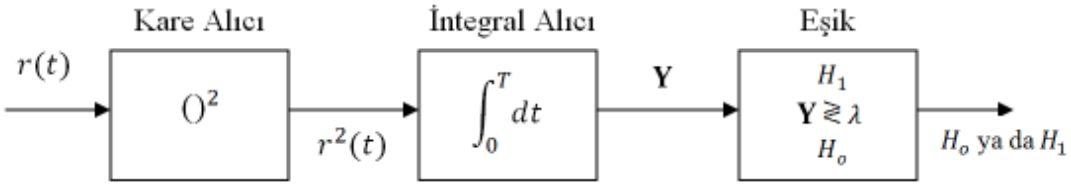
fazlasıyla sınırlıdır (Letaief vd. 2007). Ayrıca spektrum algılaması işlemi için uyumlu süzgeç kullanacak bir BR sistemi farklı tip birincil kullanıcılar için farklı süzgeçler gerektirecektir ki, bu da sistemin karmaşıklığının ve maliyetinin artmasına sebep olacaktır (Akyildiz vd. 2009).

2.1.4.2. Enerji Algılaması

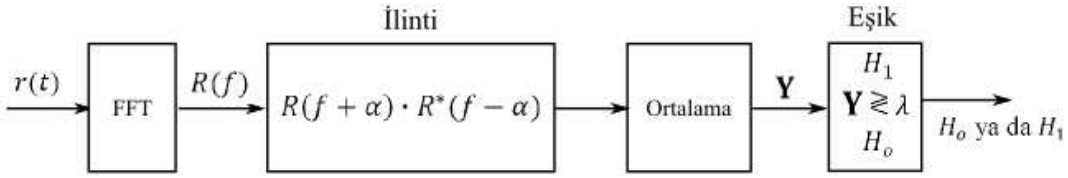
Enerji algılaması metodu, sadece Gaussian gürültünün gücünün bilindiği durum için optimaldir (Sahai vd. 2004). Bu yaklaşımda, ikincil kullanıcılar birincil kullanıcının varlığını, alınan işaretin gücüne göre tespit ederler (Şekil 2.4). Uygulaması kolay bir sistem olsa da zayıf güçlü işaretleri algılamaya çalıştığında uyumlu süzgeçlere göre daha uzun sürelerde işlem gerçekleştirir. Ayrıca algılama işlemi alınan sinyalin SNR değerine göre yapıldığından, gürültü seviyesindeki belirsizlik, performans üzerinde etkilidir. Üstelik enerji algılaması işareti saptayabilse de işaret tiplerini ayırt edemediğinden, çoğunlukla istenmeyen BR işaretleri yanlış saptamalara sebebiyet vermektedir (Akyildiz vd. 2009).

2.1.4.3. Dairesel-durağan öznitelik algılaması

Modülasyonlu işaretler genellikle sinüs taşıyıcılarla, dürtü dizileriyle ya da dairesel önadlarla eşlendiklerinden kendi içlerinde bir periyodiklik barındırırlar. Bu işaretler, özilinti fonksiyonları periyodik olduğundan dairesel-durağan olarak adlandırılırlar.



Şekil 2.4 Enerji Algılaması



Şekil 2.5 Öznitelik Algılaması

Dairesel-durağan öznitelik algılaması, gelen sinyalin spektrum ilintisinin *T* aralıklarında ortalamasının alınması ve eldeki test istatistiğiyle kıyaslanması yoluyla yapılabilir (Şekil 2.5), (Cabric vd. 2004). Bu algılama yöntemi rastgele gürültüye ve diğer modülasyonlu işaretlerden girişime karşı daha dayanıklıdır (Letaief vd. 2007).

2.1.4.4. İşbirlikçi spektrum algılama

Spektrum algılama problemini doğrudan alıcıda bazı parametrelere dayalı çözmenin yanı sıra benzer işlemi işbirlikçi yolla yapmaya çalışma işlemi olan işbirlikçi spektrum algılama yapısı da fazlasıyla ilgi görmüş bir konudur. İşbirlikçi spektrum algılama bir çift bilişsel terminalin dağınık işleme stratejisiyle çalışmasıyla gerçeklenebilir (Gandetto vd. 2007). Bu sistem terminallerin algılama problemine yaklaşımlarını birbirleriyle paylaştığı 'dağıtılmış işbirlikçi bilinç' yapısına dayanır.

İşbirlikçi spektrum algılama özellikle birincil ya da lisanslı kullanıcıların varlığını saptamada ciddi zorluklar çıkaran çok yollu sönümlenme gibi problemlerin önüne geçmek için tasarlanmıştır. Bu sistemlerde birçok ikincil kullanıcıdan alınan veriler kullanılarak daha yüksek algılama performansı elde edilmesi hedeflenmektedir (Uchiyama vd. 2008).

Bilişsel radyonun TV bantlarında lisanssız kullanımını araştırmak için Kasım 2004'te kurulan IEEE 802.22 WRAN (Kablosuz Bölgesel Alan Ağları – Wireless Regional Area Network) çalışma grubu da spektrum algılama problemi için işbirlikçi yaklaşımı savunmuştur (Sivanesan vd. 2006). Şehir-dışı alanlarda 33 ila 100 km yarıçapında bir hücreyi kapsayacak bir baz istasyonuna sahip olacak sistem, CPE (Öncül Tüketici Cihazı - Consumer Premise Equipment) olarak tanımlanan sabit konumlu ikincil kullanıcılara geniş bant Internet hizmetleri sağlayacaktır. Bu kapsamda CPE'lerin iletim

parametrelerinin kontrolünü sağlamakta kullanılabilecek radyo ortamının kesin ve güncel bilgisinin işbirlikçi algılama yöntemleri uygulanarak sağlanması gerektiği öngörülmüştür.

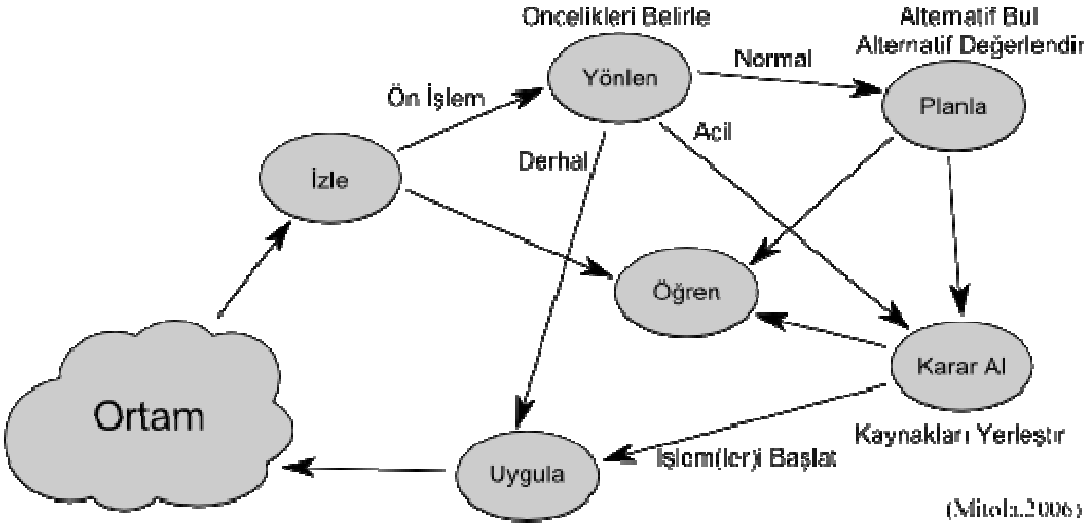
2.1.5 Bilişsellik

Farklı yöntemlerle elde edilen verilerin anlamlandırılması ve bunlardan sonuçlar çıkarılması için analiz birimi gibi iş görecek bir sistem ya da sistemler bütününe ihtiyaç duyulmaktadır. BR mimarisi içindeki bilişsel döngü; zamansal organizasyon, çıkarımlar akışı ve kontrol durumlarını kapsar (Mitola 2006). Bilişsel işlemler temel olarak şu konularla ilgilenirler (Kushwaha vd. 2008):

- Spektrum yönetimi, fırsatçı spektrum erişimi
- Optimal iletim oranı kontrolü
- QoS (Servis Kalitesi – Quality of Service)

Bilişsel döngüde CBR (Olaylara Dayalı Akıl Yürütme – Case-Based Reasoning) yapısına benzer şekilde 'olaya dayalı karar verme', 'bilgi tabanı' ve 'akıl yürütme' sistemleri kullanılabilir (Le vd. 2007). Bu sistemde bir problemler grubu, bir uygulamalar grubu ve bir sonuçlar grubu vardır. Bir olay bir problemin, bir uygulamanın ve bir sonucun çokuzlumu (tuple) olarak tanımlanır. Ayrıca hafıza bilinen olayların kümesi olarak tanımlanır. Bir algılayıcı sistem yeni bir problemle karşılaştığında bilişsel sistem yapacağı uygulamaya karar vermelidir. En iyi hareketin gerçekleştirilebilmesi için bilişsel sistem durumu hafızadakilerle karşılaştırır. Olay analizi yeni probleme en yakın ve en iyi sonucu vermiş olan eski olayı seçer. Buradaki önemli nokta olaylar arası ilişkileri tanımlayacak güçlü bir algoritmanın tanımlanmasıdır.

Spektrum yönetimi olarak incelenen sistemlerde temel amaç, algılayıcı tarafından bulunan spektrum boşlukları ve iletim gücü kontrolü çıktısı göz önünde bulundurularak, radyo ortamının zamanla değişen koşullarına uyum sağlayan ve kanal üzerinde sürekli güvenilir iletişim sağlayan bir modülasyon stratejisinin seçilmesidir (Haykin 2005).



Şekil 2.6 Basitleştirilmiş bilişsellik döngüsü

Mitola (2006), bilişselliği tüm BR yapısını kapsayacak şekilde modellemiştir (Şekil 2.6). Bu yapıyı diğer konularla örtüştürmek gerekirse "İzle" yapısı spektrum algılamasına "Uygula" yapısı da radyo çalışma parametrelerinin değiştirilmesine tekabül etmektedir. Sistemin asıl olarak bilişsel olmasını sağlayan "Öğren" yapısıyla beraber diğer işlemler spektrum yönetimi konusu altında toplanabilir. Bu döngünün bir çalışmasına uyanık dönem (wake epoch) adı verilir, çünkü yapılan akıl yürütme ortama bağlıdır. Bunun dışında sistemin kendi içinde akıl yürütme yaptığı uyuyan dönemler (sleep epoch) ve daha üst bir otoriteden karar beklediği rica dönemleri (prayer epoch) olabilir.

QoS bilişsel döngünün asıl amacını belli eden metrikler topluluğu olarak işlev alabilir. Bu durumda bilişsel sisteme terminal kontrol fonksiyonlarını ve QoS isteklerini kullanıcılardan alan bir birim dâhil edilir (Jondral 2007). Ayrıca QoS yapılacak her türlü optimizasyonda bir sınır fonksiyonu olarak yer alabilir. Söz konusu optimizasyonları bilişsel sistemin gerçekleştireceği göz önünde bulundurulursa QoS doğrudan bu sisteme bir girdi olarak tasarlanabilir.

2005'in ilk çeyreğinde yeni nesil radyo ve gelişmiş spektrum yönetimi teknolojilerini standardize etmek için kurulan IEEE P1900 Standartlar Komitesi, Mart 2007'de Standart Koordinasyon Komitesi 41 (Standards Coordination Committee 41 – SCC41)

olarak tekrar organize edilmiş ve IEEE 1900.x şeklinde çalışma grupları olarak araştırmalarına devam etmektedir (Pawelczak vd. 2006).

2.1.6 Yazılım Tanımlı Radyo

Algılanan durumlar ve bilişsel sistem tarafından alınan kararların işletilebilmesi kullanılan altyapının bu kararları uygulayabilmesi gerekmektedir. Bunun gerçekleşebilmesi için altyapı çalışma parametrelerini bir şekilde düzenleyebiliyor olmalıdır. YTR bu problem için bir çözüm sunmaktadır.

BR, YTR ve YR sıklıkla karıştırılan birbirine yakın ama farklı kavramlar içeren terimlerdir. YR radyo çalışma parametrelerinin yazılım tarafından belirlendiği iletişim sistemi olarak tanımlanabilirken YTR bu işlemin pratik olarak uygulanabilirliğini sağlayan donanımsal elemandır (Mitola 2000). BR ise parametreleri düzenleyecek yazılımın kontrolünün çalışma koşullarına bağlı olarak bilişsel süreçle belirlenmesinden bahseder. Terimler çalışma alanlarının yanı sıra Joseph Mitola tarafından tanımlanmış olma özelliğini de paylaşırlar. YTR hakkındaki radyo sistemlerindeki yeni devrim olarak nitelediği çalışmasını 1990'lı yılların başında hazırlayan Mitola, sonraki yıllarda öncelikle askeri bir projede bir YTR uygulaması gerçekleştirmiştir. Daha sonra YTR sistemlerinin bir mimariye ihtiyacı olduğunu düşünen Mitola YR mimarisini tanımlamış ve 1990'lı yılların sonlarında da bu mimari üzerinde çalışabileceğini düşündüğü BR yapısını tanımlamıştır.

İdeal bir YR anten çıktısını direkt olarak örnekler (Jondral vd. 2007). YTR ise gerçeklenebilir bir yapı olarak, alınan sinyalleri uygun bir bant-seçen süzgeç ardından örnekler. Çalıştığı alana bağlı olarak bir YTR aşağıdakilerden herhangi biri olabilir:

- Bir kablosuz standarda özgü birden fazla frekans bandını destekleyen çok bantlı sistem (örn. GSM 900, GSM 1800, GSM 1900).
- Birden çok hava ara yüzünü destekleyen birden çok standardı destekleyebilen sistem. Çok standartlı sistemler tek bir standart ailesinde (örn. UTRA-FDD, UTRA-TDD for

UMTS) ya da farklı şebekeler arasında (örn. DECT, GSM, UMTS, WLAN) çalışabilir.

- Farklı servisler sağlayan birden çok servisi barındırabilen sistem (örn. telefon, veri, video).
- İki ya da daha fazla bağımsız gönderme ve alma kanalını aynı anda destekleyebilen çok kanallı sistem.

YTR daha basit bir anlatımla ses kartı bulunan herhangi bir bilgisayar olarak düşünülebilir. Böyle bir yapıda ses kartı, ara yüzleri vasıtasıyla aldığı analog sinyali sayısala çeviren bir sistem, bilgisayar işlemcisi de yazılım çalıştırabilen bir birim olarak görülebilir. Bu sayede ses kartının örnekleyebileceği frekans aralığı ve bilgisayar işlemci gücü ile sınırlı bir YTR yapısı elde edilmiş olur.

Herhangi bir bilgisayarı doğrudan bir YTR sistemine çevirmek gibi bir amacı olmasa da en azından üzerinde çalışmak için bir ortam sağlayabilen GNURadio temel anlamda ucuz RF donanımı ile YR altyapısı kurmayı amaçlayan açık kaynak kodlu bir yazılım kütüphanesidir (GNU). Ayrıca savunma sanayisinde ileri seviye işler için farklı YTR uygulamaları bulunmaktadır.

2.2 Yapay Sinir Ağları

Yapay sinir ağları fikri, insan beyninin şu anki sayısal bilgisayarlardan farklı çalışmasına rağmen örüntü tanıma, algılama ve motor kontrol gibi işlemlerde çok daha hızlı sonuçlar elde ettiği gerçeğinden yola çıkmıştır (Haykin 1999). Beyin, nöron denilen yapısal parçacıkları organize ederek birçok işlemi en hızlı bilgisayarlardan bile çok daha etkin bir şekilde gerçekleştirebilmektedir. Ortalama bir insan beyninde $3x10^{10}$ nöron bulunmakta ve her biri 10^4 e kadar nörona bağlanabilmektedir. Genellikle elektronik bileşenlerden oluşan ya da özel yazılımlarla bilgisayar ortamında benzetim yapılan yapay sinir ağları da yapay nöronlardan oluşarak beynin bazı işlemleri gerçekleştirmede kullandığı yapıyı modellemeye çalışır. Bu doğrultuda bir YSA, çalışması için gerekli bilgiyi bulunduğu ortamdan bir öğrenme işlemiyle alır ve bu bilgiyi sinaptik ağırlık adı verilen nöronlar arası bağlantılarda saklar.

YSA'nın çalışma birimi olan yapay nöron, uçlarından aldığı girdiyi işleyip aktivasyon fonksiyonu adı verilen bir transfer fonksiyonuna göre çıktı üretir (Du vd. 2006). Bu fonksiyon girdiyi çıktıya lineer ya da lineer olmayan bir şekilde eşlemler ve $\phi(\cdot)$ ile gösterilir. Şekil 2.7'de gösterilen McCulloch–Pitts nöron modeli biyolojik yapıya benzer olmasıyla birçok YSA uygulamasında kabul görmüştür. Bu yapıdaki bir nöronun çıktısı;

$$net = \sum_{i=1}^{J_1} w_i x_i - \theta = \mathbf{w}^{\mathrm{T}}\mathbf{x} - \theta \tag{2.1}$$

$$y = \phi(net) \tag{2.2}$$

olarak ifade edilir. x_i, girdileri w_i ağırlıkları ifade eder. θ ise, kutuplama ya da eşik değeridir. $\Phi(.)$, ile gösterilen aktivasyon fonksiyonu ise genellikle herhangi bir reel değeri (-1,1) ya da (0,1) aralığına eşleyen sürekli ya da sürekli olmayan bir fonksiyondur. Bu fonksiyon aşağıdakilerden bazıları olabilir:

- Eşik fonksiyonu

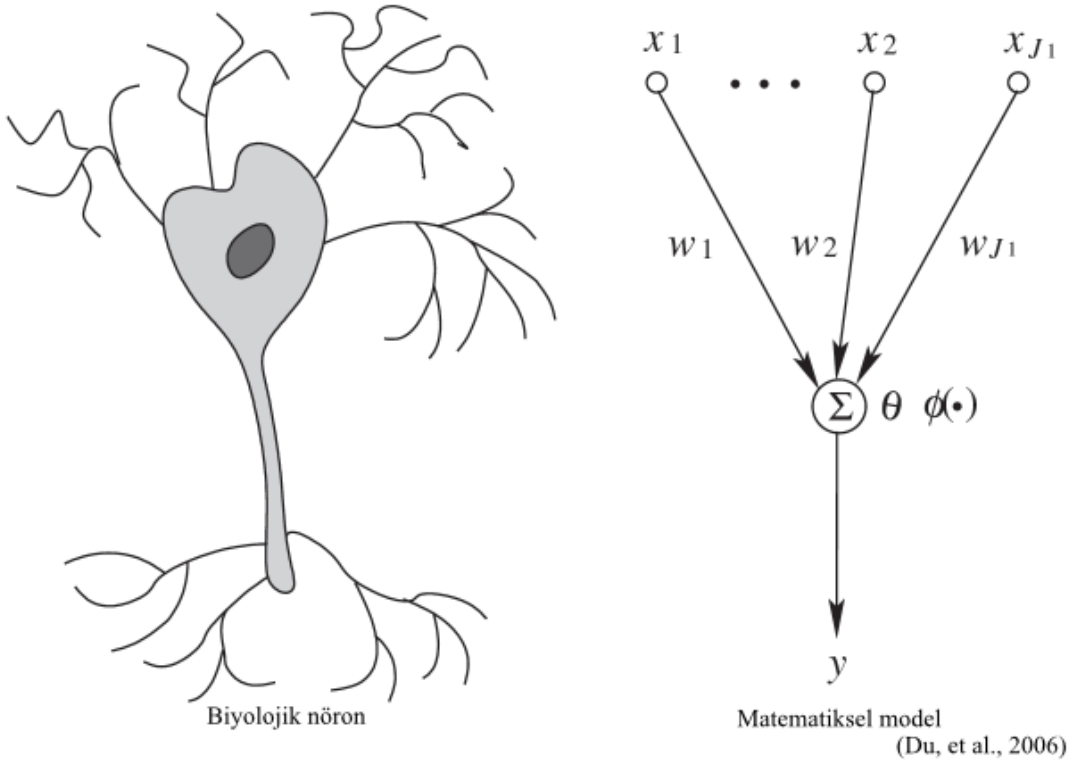
$$\phi(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{2.3}$$

- Yarı-lineer fonksiyon

$$\phi(x) = \begin{cases} 1, & x > a \\ \frac{x}{2a} + \frac{1}{2}, & -a \leq x < a \\ 0, & x < -a \end{cases} \tag{2.4}$$

- Log-fonksiyon

$$\phi(x) = \frac{1}{1+e^{-\beta x}} \tag{2.5}$$



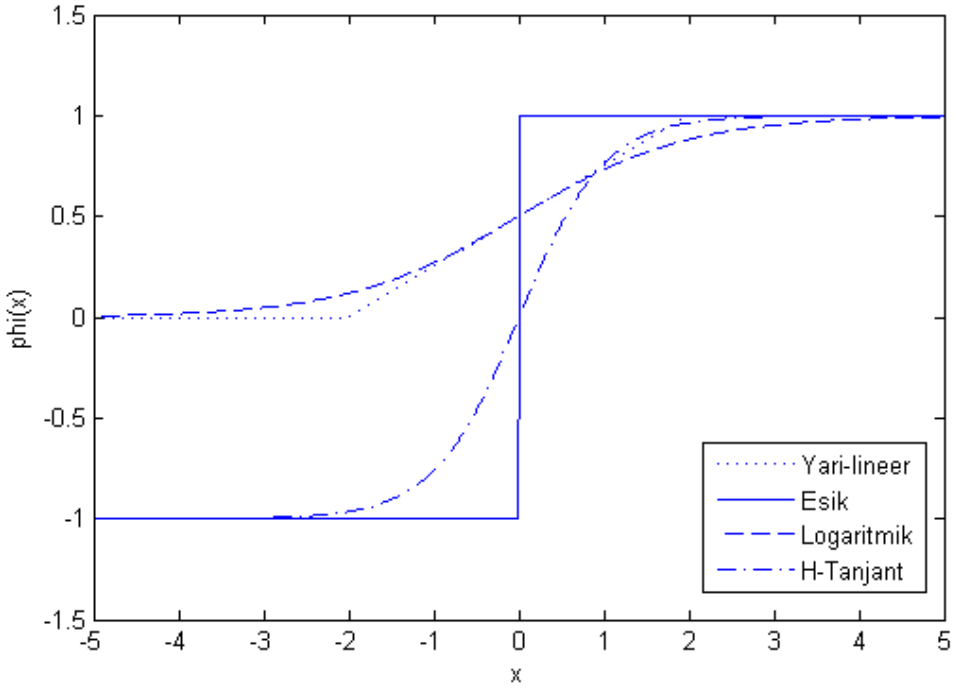
Şekil 2.7 McCulloch–Pitts nöron modeli

- Hiperbolik tanjant fonksiyonu

(2.6)

Bu fonksiyonlardaki bir kazanç değeri olup genellikle birim seçilir ve aktivasyon fonksiyonunun dikliğini belirler. Fonksiyonların çıktıları Şekil 2.8'de gösterilmiştir.

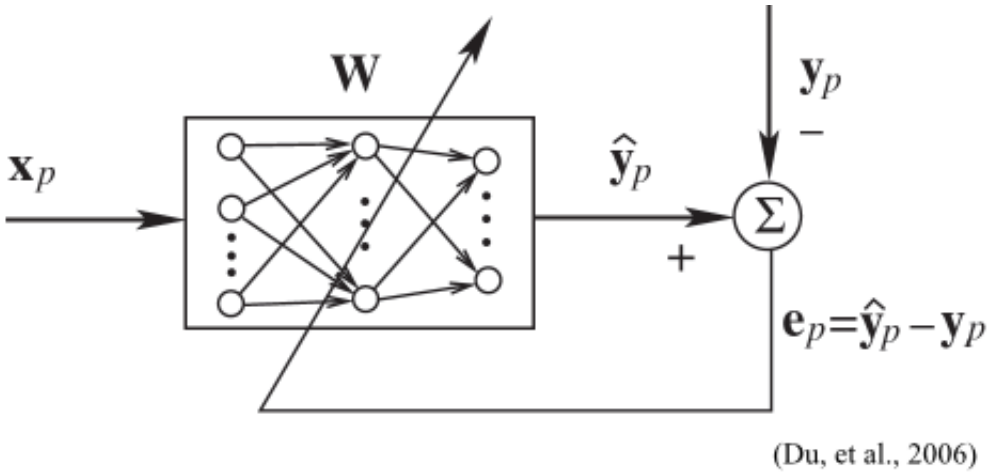
, ağırlık matrisi, değeri ucundan ucuna olan bağlantıdaki ağırlığı göstermek üzere bir YSA'nın mimarisini tanımlar (Du vd. 2006). olduğu durumda ile arasında bir bağlantı olmadığı anlaşılır. Bu ağırlık değerlerine bakarak YSA'lar temelde ileri beslemeli ağlar ve özyinelemeli ağlar olarak gruplanabilir. İleri beslemeli ağlar genellikle katmanlar halinde tasarlanır ve nöronlar arası bağlantılar sadece bir katmandan sonrakine doğru olur. Nöronlar kendi bulundukları katmana ya da bir önceki katmana bağlanmazlar. Bir özel durum olarak bir katmandaki tüm uçlar sonraki katmandaki tüm uçların hepsine bağlıysa bu YSA'ya tam bağlantılı ileri beslemeli ağ denir. Bu yapıdaki popüler ağ yapıları MLP (Çok-Katmanlı Perceptron –



Şekil 2.8 Aktivasyon Fonksiyonları

Multi-Layer Perceptron) ve RBFN (Radyal Taban Fonksiyon Ağı – Radial Basis Function Network) mimarilerini içerir. Özyinelemeli bir ağ ise en az bir geri besleme bağlantısı içermesi açısından ileri beslemeli ağlardan ayrılır. Geri besleme bağlantısından kasıt, bir ucun çıktısının bir gecikme ya da integral fonksiyonu sonrasında kendi girişine bağlanmasıdır. Popüler özyineli ağ yapıları Hopfield ağları ve Boltzmann makinesini içerir. Birçok YSA, girdiyi doğru bir çıktıya eşleyecek şekilde örüntü ilişkilendirici olarak çalışır (Arbib 2003). Gerek ileri beslemeli gerekse özyinelemeli ağlar için öğrenme kuralları da bu örüntüler arasında kullanışlı bağlantılar oluşturacak şekilde tasarlanır.

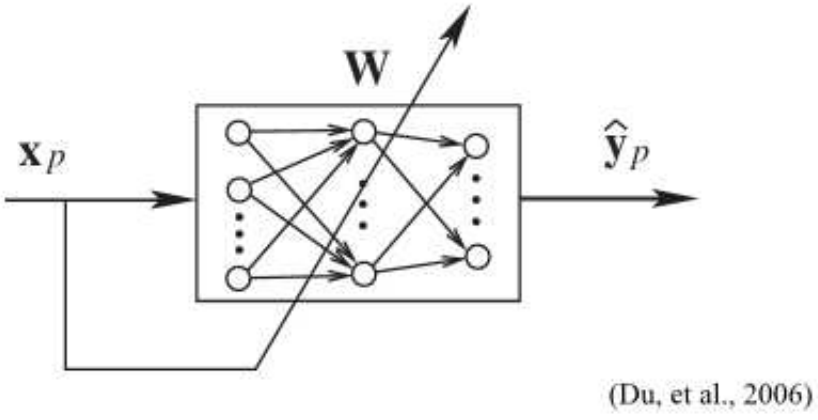
Öğrenme yapıları iki temel modele sahiptir: Öğreticili öğrenme ve öğreticisiz öğrenme (Haykin 1999). Kavramsal olarak bir öğreticinin, YSA'nın çalışacağı ortamın bilgisine girdi-çıktı örnekleri şeklinde sahip olduğu düşünülebilir. Bu durumda öğretici, ortamdan alınan herhangi bir eğitim örneği için YSA'ya beklenen çıktıyı sağlayabilecektir. Ağın çalışma parametreleri, eğitim örneği ve beklenen çıktı ile oluşan çıktı arasındaki hata sayesinde düzenlenebilir. Bu tip bir eğitimin performansı eğitim örneğinin ortalama



Şekil 2.10 Öğreticili Öğrenme

karesel hatası ile ölçülebilir. Sistemin zamanla performansının iyileşmesi için bu değerin hata yüzeyinde bir minimum noktaya ulaşması gerekir (Şekil 2.10).

Öğreticisiz öğrenmede ise beklenen değerleri doğrudan sağlayabilecek bir yapı mevcut değildir. Onun yerine ağ parametreleri, yapılan işten bağımsız bir gösterim kalitesi değeri ile düzenlenir. Ağ girdi verisinin istatistik düzenine göre kendini düzenler ve girdideki özniteliklerin bir gösterimini oluşturur (Şekil 2.10). Bazı yerlerde öğreticili, bazı yerlerde ise öğreticisiz öğrenme konusu dâhilinde yer alan ve 2.3'te genişçe yer verilen destekleyici öğrenme konusunda ise ağ, girdi-çıktı arasındaki bağıntıyı, ortamıyla sürekli haberleşerek bir hata değerini küçültmek suretiyle öğrenir. Bu öğrenmede değerlendirici (critic) yapı ortamdan aldığı destekleyici değerleri yorumlar



Şekil 2.9 Öğreticisiz Öğrenme

ve öğrenme yapısına iletir.

Bu temel modeller haricinde öğrenme yapıları işlem görme hallerine göre de gruplanabilmektedir. Bunlar, hata-düzeltme öğrenmesi gibi genel algoritmalardan benzetimli tavlamaya (simulated annealing) ve evrimsel algoritmalara kadar birçok konuyu kapsayabilmektedir (Arbib 2003).

2.3 Destekleyici Öğrenme

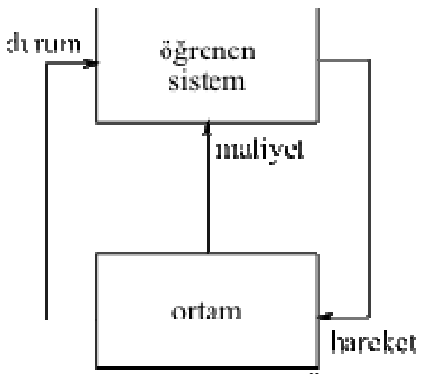
Destekleyici (reinforcement) terimi deneysel psikolojide hayvanların öğrenme konusundaki çalışmalardan gelmektedir, bir tepki oluşturan ve aynı durumda yine aynı tepkiyi verme olasılığını arttıran bir olayın oluşması anlamındadır (Barto 2003). Psikologlar tarafından kullanılmayan 'destekleyici öğrenme' ise yapay zekâ ve mühendislik alanındaki araştırmacılar tarafından bu prensip üzerinde çalışan öğrenme yapılarını ifade etmekte kullanılır. Destekleyici öğrenme en basit haliyle, bir hareketin tatmin edici koşullar oluşturması ya da koşulları iyileştirmesi durumunda bu hareketin yapılması eğiliminin artırılması ya da bu hareketin 'desteklenmesi' olarak tanımlanır. Kavram, 1950'lerden beri var olsa da konu üzerindeki araştırmalar; karmaşık ortamlarda çalışacak otonom robotların tasarlanması ve çok büyük dinamik karar alma problemleri için kullanışlı yaklaşımlar üretilmesi gerekliliklerileyle hızlanmıştır. Destekleyici öğrenme, en iyi hareketleri üretecek stratejiyi bulmaya yarayan bir optimizasyon yöntemi olarak adlandırılsa da pratikte genellikle en iyi stratejiyi bulmaktansa sistemin sürekli iyileşmesi daha önemlidir.

Destekleyici öğrenme, “klasik yaklaşım” ve “modern yaklaşım” şeklinde iki temel üzerinden araştırılır (Haykin 1999). Klasik yaklaşımda öğrenme işlemi yüksek performanslı bir strateji elde etmek adına sistemi ödüllendirmek ve cezalandırmak şeklinde ilerler. Modern yaklaşım ise dinamik programlama kavramı dâhilinde planlama adına sonraki muhtemel hareketleri de düşünerek karar vermeyi amaçlar. Bu yapıda yapılacak her hareketin o an için bir maliyeti ve o hareketin sonucunda ileride yapılabilecek hareketlerin maliyeti beraber düşünülür. Dinamik programlama, anlık maliyetlerle gelecekteki maliyetler arası ödünleşimi (tradeoff) hesaplamaya çalışır (Bertsekas vd. 1996).

Destekleyici öğrenme, tüm yapıyı bir ortam ve onunla haberleşen bir öğrenme sistemi olarak modeller (Şekil 2.11), (Barto 2003). Öğrenme sistemi ortamla, sonlu ya da sonsuz olabilen $n = 1,2,3,\ldots$ ayrık zaman adımlarında etkileşimde bulunur. Her n zaman adımında öğrenme sistemi ortamın o anki durumunu, X_n, alır ve buna dayanarak bir hareket, a_n, gerçekleştirir. Buradaki X_n'ler ortam durumlarının kümesini, a_n de X_n durumunda yapılabilecek hareketlerin kümesini oluşturur. Bir zaman adımı sonrasında öğrenme sistemi reel bir maliyet değeri, , alır ve durumuna geçer. Bu maliyet değeri sadece öğrenme sisteminin hareketiyle değil, bulunulan durumuyla ve diğer faktörlerle belirlenebilir. Genel olarak bu maliyet fonksiyonu şu şekilde gösterilir (Haykin 1999):

(2.7)

Burada yapılması gereken durumları hareketlere eşleyecek bir kuralın oluşturulmasıdır.



Şekil 2.11 Destekleyici Öğrenme

Öğrenme sistemi bu kuralı kullanarak ortamın o anki durumunda hangi hareketi yapması gerektiğini bulacaktır.

$$\pi = \{\mu_0, \mu_1, \mu_2, \dots\} \tag{2.8}$$

Genellikle π ile gösterilen kural, $n = 0,1,2, \dots$ zaman adımlarında $X_n = i$ durumunu $A_n = a$ hareketine eşleyen μ_n fonksiyonlarından oluşur. Bu kural zamanla değişen ya da değişmeyen yapıda olabilir. Zamanla değişmeyen bir kuralı

$$\pi = \{\mu, \mu, \mu, \dots\} \tag{2.9}$$

olarak ifade etmek daha doğru olacaktır. Destekleyici öğrenme sistemi, bu kuralı, zaman içerisinde oluşacak maliyeti küçültecek şekilde düzenlemeye çalışır. $X_0 = i$, durumundan başlayan ve $\pi = \{\mu_n\}$ kuralını kullanan bir sonsuz-ufuk (infinite-horizon) probleminde toplam beklenen maliyet;

$$J^{\pi}(i) = E[\sum_{n=0}^{\infty} \gamma^n g(X_n, \mu_n(X_n), X_{n+1}) \mid X_0 = i] \tag{2.10}$$

olarak tanımlanır (Haykin 1999). Buradaki $J^{\pi}(i)$, i durumundan başlayan π kuralı için ilerleme maliyeti (cost-to-go) fonksiyonu, $\gamma \in [0,1)$ ise indirim faktörü (discount factor), adını alır. γ değeri düzenlenerek sistemin hareketlerinin kısa ya da uzun dönemdeki sonuçlarıyla ilgilenmesi sağlanabilir. Örneğin $\gamma = 0$ limit değerinde sistem sadece anlık sonuçlarla ilgilenecek ileride oluşacak maliyetleri önemsemeyecektir. İlerleme maliyeti fonksiyonunun optimal değeri ise

$$J^*(i) = \min_{\pi} J^{\pi}(i) \tag{2.11}$$

ile ifade edilir. Kural zamanla değişmeyen yapıdaysa $J^{\pi}(i)$ yerine $J^{\mu}(i)$ kullanılabilir ve eğer tüm i başlangıç durumları için $J^{\mu}(i) = J^*(i)$ ise μ optimaldir denir.

Destekleyici öğrenme probleminin bu modeli, MDP (Markovian Decision Process – Markov Karar Süreci) teorisi üzerinden tasarlanmıştır (Barto 2003). MDP, çalıştığı ortamın Markov özelliği göstermesini gerektirmektedir. Bu da sistemin i durumundan a_{ik} hareketiyle j durumuna geçme olasılığının sadece ve sadece i durumuna bağlı olduğu anlamındadır (Haykin 1999). Dolayısıyla tam bir MDP tanımlaması, durum geçişlerinin ve maliyetlerin, durumlarla ve hareketlerle nasıl değiştiğinin olasılık detaylarını içerir. Teorik olarak bir MDP'nin optimal kurala ulaşma amacı çeşitli dinamik programlama algoritmalarıyla halledilebilir, ancak bu algoritmaların çözümsel karmaşıklığı büyük ölçekli problemler için pratiklikten uzaktır. Destekleyici öğrenme temelde MDP ile benzer probleme yoğunlaşsa da, optimal davranışı olasılık modelleri üzerinden çalışma öncesi hesaplama yerine, çalışma zamanında hesaplaması açısından farklılaşır.

Bir destekleyici öğrenme sistemi şu özelliklere sahiptir (Barto 2003):

1. Ortam ve öğrenme sistemi tam bir kesinlik içermemektedir, dolayısıyla yapılan hareketlerin oluşturacağı çıktılar önceden kesin bilinemez. Bu kesinlik eksikliğini oluşturan rastgele değişikliklerin bir olasılık modeli olabilir ya da olmayabilir.
2. Maliyet değeri öğrenme sisteminin hareketlerini değerlendiren herhangi bir değer olabilir. Bu değer hedef duruma geçildiğini ya da geçilemediğini ifade eden bir belirleyici olabildiği gibi öğrenme sisteminin davranışlarını sürekli değerlendiren bir değer de olabilir. Hatta birçok kıstas birleştirilerek tek bir maliyet değeri elde edilebilir.
3. Destekleyici öğrenme sistemi genellikle anlık düşük maliyet yerine ileride daha düşük maliyet sağlayacak bir hareket seçer.
4. Maliyet değeri hangi hareketin en iyisi olduğunu ya da hareketlerini ne şekilde değiştirmesi gerektiğini belirlemez. Bu destekleyici öğrenmeyi, öğreticili öğrenmeden ayıran en önemli özelliktir.
5. Destekleyici öğrenme bir seçim sürecidir. Dolayısıyla en iyi sonucu üretmek üzere aralarından seçim yapılabilecek çeşitli hareketler bulunmalıdır. Bu çeşitlilik açınsama (exploration) özelliğini beraberinde getirir.

6. Sistem açınsama ve kullanma (exploitation) hareketleri arasında dengelenmelidir. Öğrenme sistemi daha önce öğrendiklerini kullanarak daha düşük maliyetler elde etmeli aynı zamanda açınsayarak yeni yollar öğrenmelidir. Destekleyici öğrenmeyi hem öğreticisiz öğrenme hem de öğreticili öğrenmeden ayıran özellik budur. Açınsama ve kullanma davranışsal yapıları ikisinde de mevcut değildir.

Maliyet fonksiyonu olarak yukarıda kullanılan ilerleme maliyeti fonksiyonu dışında kullanılabilecek diğer değer de Q değeridir.

$$Q^{\mu}(i,a) = c(i,a) + \gamma \sum_{j=1}^{N} p_{ij}(a) J^{\mu}(j) \tag{2.12}$$

$$c(i,a) = \sum_{j=1}^{N} p_{ij}\, g(i,a,j) \tag{2.13}$$

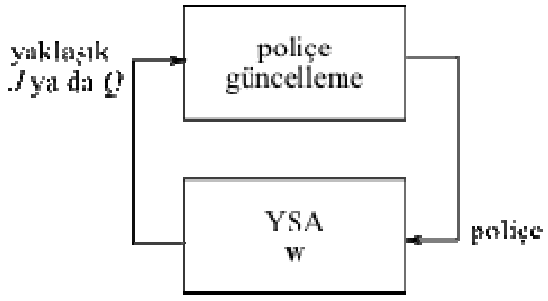
Buradaki p_{ij} değeri i durumundan j durumuna geçiş ihtimalini gösterirken $p_{ij}(a)$ ise, i durumundan j durumuna a hareketiyle geçiş ihtimalini göstermektedir. N ise sistemdeki tüm durumların sayısıdır. Hem J hem de Q değerinin kullanışlı olmasının sebebi sistemin bir MDP olmasından kaynaklıdır (Barto 2003). J^* ya da Q^* fonksiyonunun bilinmesi durumunda sistemin herhangi bir durumunda yapılacak en uygun hareket hesaplanabilir. Örneğin i durumunda yapılacak en uygun hareket $Q^*(i,a)$ değerini en küçükleyen herhangi bir harekettir. Bir π kuralına bağlı karar sağlayan destekleyici öğrenme sistemi bu değerler sayesinde kuralını sürekli olarak iyileştirebilir. Bunun anlamı öğrenme sistemi sadece π'ye bağlı kalarak karar vermek yerine J^{π} değeri ile hesaplama yaparsa en fazla π kuralının üreteceği maliyet kadar elde edecektir ve daha düşük bir değer oluşturması da ihtimal dâhilindedir. Ancak sistemin tam bir Markov modelinin oluşturulamadığı ya da durum uzayının hesaplamalar için çok büyük olduğu yapılarda optimal değeri aramak yerine yaklaşım yapmak daha geçerli bir çözüm olmaktadır.

Bu doğrultuda, Monte Carlo benzetimiyle kural döngüsüne yaklaşım yapılabileceği önerilmiştir (Bertsekas vd. 1996). Bu algoritma verilen bir durumu için değeri ya da değeri için sırasıyla ve yaklaşım yapan bir fonksiyon kullanılır. Buradaki değeri optimizasyon yoluyla tanımlanan bir parametre vektörüdür ve uygulamasında bir YSA kullanılabilir. Bu uygulamada değeri YSA'nın sinaptik ağırlıklarını gösterecek şekilde olarak gösterilebilir (Şekil 2.12). Algoritma, 'yaklaşık kural değerlendirme adımı' ve 'kural iyileştirme adımı' arasında değişerek ilerler (Haykin 1999). Değerlendirme adımında gerçek değer fonksiyonuna yaklaşım yapan ya da fonksiyonu tüm durumları için hesaplanır. İyileştirme adımında ise yaklaşım fonksiyonu kullanılarak iyileştirilmiş kuralı oluşturulur.

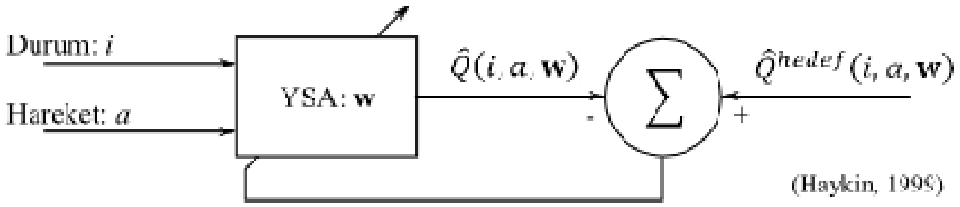
2.4 Q-öğrenme

Destekleyici öğrenmenin temel amacı olan optimal bir kuralın çeşitli hareket dizilerinin denenmesiyle bulunması olduğu düşünülürse, bu işlem sadece deneyimlerden yola çıkarak çalışma anında da yapılabilir (Haykin 2005). Q-öğrenme, optimal kuralı adım adım ilerleyerek bulan artırımlı bir dinamik programlama yöntemidir. Geçiş olasılıklarının tam olarak bilinmediği MDP ortamları için gayet uygun olan ancak tüm ortamın izlenebilir olmasını gerektiren bir yapıdır.

Yukarıda formülü verilen Q değeri, durumunda hareketini gerçekleştirip kuralıyla devam edildiğinde oluşacak maliyetin indirimli beklenen değeri olarak tanımlanabilir (Watkins vd. 1992). Q-öğrenmenin amacı da optimal bir kuralın Q değerlerini tahmin etmektir. Optimal ilerleme maliyeti için $J^*(i)=min_a Q^*(i,a)$ yazılabileceğinden a^*



Şekil 2.12 Yaklaşık kural döngüsü



Şekil 2.13 Hedef Q değerine YSA ile yaklaşma

minimum maliyetin elde edildiği hareket olarak gösterilirse optimal kural $\pi^*(i)=a^*$ olarak tanımlanabilir. Buradan hareketle eğer öğrenme sistemi Q değerlerini öğrenebilirse optimal hareketi de bulabileceği söylenebilir. Q-öğrenmedeki öğrenme sisteminin deneyimleri bölümler halinde gösterilebilir. Bir bölümünde sistem; durumunu gözlemler, hareketi seçer ve uygular, sonraki durumunu gözlemler, maliyeti alır ve Q değerlerini bir öğrenme faktörüne göre şu formülle düzenler.

(2.14)

Watkins (1992), yukarıda tek bir döngüsü verilen algoritmanın ilk bölümlerinde Q değerlerinin tanımladıkları kuralı tam olarak yansıtmayacağını ancak döngüler devam ettikçe yakınsadığını göstermiştir. Ancak bu yapı Q değerlerinin bir tabloda tutulduğunu kabul etmektedir. Değerlerin saklanması için bir tablo kullanımı etkin bir yöntem olsa da durum-hareket çiftlerinin çok sayıda olduğu durumlarda sistem, gerektireceği büyük miktarda hafızadan ötürü maliyetli olabilir.

Q değeri güncelleme formülü;

(2.15)

şeklinde yazıldığında köşeli parantez içindeki kısım bir hata sinyali olarak düşünülürse, bölümündeki hedef Q değeri;

(2.16)

olarak gösterilebilir. Buradan hareketle Q-öğrenme algoritması şu şekilde gösterilebilir.

$$Q_{n+1}(i,a) = Q_n(i,a) + \Delta Q_n(i,a) \tag{2.17}$$

$$\Delta Q_n(i,a) = \begin{cases} \alpha_n(Q_n^{hedef}(i,a) - Q_n(i,a)) & ,(i,a) = (i_n,a_n) \\ 0 & ,(i,a) \neq (i_n,a_n) \end{cases} \tag{2.18}$$

Bir *n* bölümü için, **w** parametreli YSA'ya durum-hareket çifti, (i_n,a_n), verilerek $\hat{Q}_n(i_n,a_n,\mathbf{w})$ şeklindeki Q değeri yaklaşımı elde edilebilir (Şekil 2.13). Algoritmanın her döngüsünde **w** parametreleri, $\hat{Q}_n(i_n,a_n,\mathbf{w})$ değerini $Q_n^{hedef}(i_n,a_n)$ değerine yaklaştıracak şekilde düzenlenir. Ancak ağırlık vektörü değiştirildiğinde hedef Q değeri de bundan dolaylı olarak etkilenir ve bundan ötürü iki Q değeri arasındaki farkın azalacağının garantisi yoktur (Haykin 1999). Bu yüzden yaklaşık Q-öğrenme algoritması ıraksayabilir. Eğer algoritma ıraksamazsa, parametre vektörünün yaklaşık Q değerinin eğitilmiş YSA üzerinde saklandığı söylenebilir.

3. MATERYAL VE YÖNTEM

3.1 Materyal

Yakın zamandaki çalışmaların çoğu genellikle çeşitli spektrum algılama yöntemlerini araştırmış, toplam bir BR sistemindeki gerekli olacak diğer parçalar için ise sadece genel bir fikir ortaya koymuştur. BR konusunu tümden ele alan çalışmalarda spektrum yönetimi ya da sistemin bilişselliğini sağlayacak kısım için ortak kanı, bir çeşit makine öğrenmesi sistemi içermesi gerektiğidir. Gerekli öğrenmeyi sağlayacak sistemin genetik algoritmalar, yapay sinir ağları ya da uzman sistemler gibi yapılardan oluşabileceği ifade edilmiş olsa da tam bir uygulama konusunda yapılmış çalışmalar sınırlıdır.

Spektrum yönetimi problemine ikincil kullanıcıların spektrum boşluklarına yerleştirilmesi işlemi olarak bakmak kısmen doğru olsa da, sadece bu probleme odaklanan sistemin, ölçeklenme problemleri yaşaması muhtemeldir. İkincil kullanıcıları yerleştirecek yapı, sonradan sisteme eklenmesi gerekebilecek bileşenler için uygun yerleri belli etmeli ve onlarla haberleşeceği ara yüzleri tanımlamalıdır. Bu görüşle zıt olacak şekilde, tez araştırmasının ilk aşamalarında, ikincil kullanıcıların spektrum boşluklarına atanması işleminin tek bir yapay sinir ağıyla gerçekleştirilebileceği konusu araştırılmıştır. Elde edilen sonuçlar sadece bu işlem için tatmin edici olsa da mümkün olması çok muhtemel değişiklikler için sistemin bütününün tekrardan düzenlenmesi gerekeceği saptanmıştır. Dolayısıyla geçerli bir alternatif olmayacağı düşünülüp çalışmanın şimdiki haline şekillendirilmesine karar verilmiştir.

Bu çalışma, toplam iletişim kanalı sayısının belli olduğu durumda değişken sayıda ikincil kullanıcının kanal yerleşimini sağlayabilecek sistemi modellemiştir. Kanal kontrol bilgilerinin BR elemanları arasındaki iletişimi için ortak bir kontrol kanalının kullanıldığı varsayılmıştır. Spektrum algılayıcı bileşen, tasarlanan model dışında bırakılmış ve temel bir veri yolu üzerinden gelecek spektrum boşlukları bilgisinin doğru olduğu kabul edilmiştir. Model temel olarak, gelen spektrum boşlukları bilgisinin bulanık olması durumunu desteklemektedir, fakat çalışma bu bilginin kesin olduğu duruma yoğunlaşmıştır. Bilginin bulanık olduğu durumda yapılması gereken

düzenlemelere değinilmiş ancak gerekli algoritma tasarlanmamıştır. Benzetimler için spektrum algılama verisi, tüm iletişim kanalları üzerinde bazılarının dolu bazılarının boş olacağı şekilde düzgün dağılımlı rastgele değişken üzerinden üretilmiştir.

Çalışma doğrudan spektrum boşluklarının ikincil kullanıcılara atanması yerine dolaylı bir yol tercih etmiştir. Spektrum boşluklarının tersi olarak nitelendirilebilecek, birincil kullanıcıların o an için kullandığı kanallar, tüm kanal atama olasılıkları içinden elenmiş böylece ikincil kullanıcıların birincil kullanıcıları rahatsız etmeden alabileceği kanallar, geçiş yapılabilecek durumlar olarak elde edilmiştir. Tüm bu bilginin BR kapsamında merkezi bir otorite tarafından yönetildiği yapı araştırılmış olup, hesapların kullanıcılar arasında yapılması gerektiği, merkezi birimin olmadığı durum kapsam dışı tutulmuştur. Önerilen sistemin özel amaca yönelik ağlarda (ad-hoc network) çalışabilmesi, ağ kullanıcılarının belli bir oturumda kendi aralarından birini merkezi karar birimi olarak seçmesiyle mümkün olabilir. Bu sayede tüm kullanıcılar kanal yerleşimi için gerekli bilgilerini bu kullanıcıya iletecek ve bu kullanıcı da diğerlerinin yapması gereken düzenlemeleri hesaplayabilecektir.

Söz konusu sistem, MATLAB ortamında tasarlanmış ve benzetimleri yine aynı ortamda gerçekleştirilmiştir. Daha önce bahsedilen birincil kullanıcıların durumu bilgisi temel fonksiyonlar yardımıyla üretilmiş, diğer bloklar ise yine temel fonksiyonlar ile kodlanmıştır. Yapay sinir ağının eğitimi için Neural Network Toolbox içindeki fonksiyonlardan ve objelerden faydalanılmıştır.

3.2 Yöntem

3.2.1 BR sistemi başlangıç durumları

Pratik uygulamada, önerilen sistemin toplam BR sistemine dâhil olduğu anda birincil ve ikincil kullanıcıların bazı kanalları zaten kullanıyor olacağı düşünülmüştür. Benzetimler için ise böyle bir durum söz konusu olamayacağından testler, farklı başlangıç senaryoları ile tekrarlanmıştır.

3.2.2 Yapay sinir ağının başlatılması

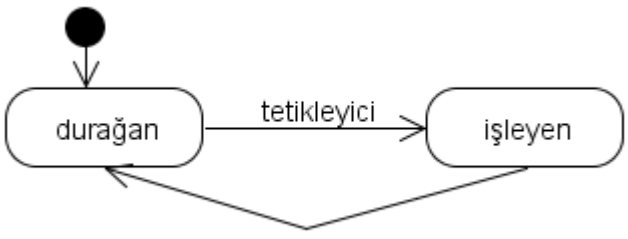
Sistem sürekli surette çalışmak üzere tasarlandığından yapay sinir ağı üzerindeki ağırlık değerlerinin de sürekli güncelleneceği, dolayısıyla sonuçların buna bağlı iyi kalacağı düşünülmüştür. Bu, tasarlanan sistemin pratik uygulamasında ağırlıkların el ile belli bazı değerlere getirilmesini gerektirmektedir. Gerekli ağırlıklar, buradaki benzetim yapısının çalıştırılması suretiyle elde edilebilir.

Bu çalışmada gerçekleştirilen benzetimde ise böyle bir başlangıç işlemi uygulanmamıştır. Bunun sebebi ise yapay sinir ağının hiç düzenlenmemiş halinde verdiği sonuçlar ile çalışma devam ettikçe verdiği sonuçlar arasında kıyas yapabilmeyi sağlamaktır. Böylece yapay sinir ağının beklendiği gibi çalışıp çalışmadığı tartışılacaktır.

3.2.3 Sistemi tetikleyici faktörler

Tüm yapı, bir durağan durumu bir de işleyen durumu olacak şekilde düşünülmüştür (Şekil 3.1). Başlangıçta her zaman için durağan durumdan başlayacak yapı işleyen duruma geçmek için bazı tetikleyicilere ihtiyaç duymaktadır. Bunlar;

- Birincil kullanıcıların durumunun değişmesi,
- İkincil kullanıcıların ihtiyaçlarının değişmesi,
- BR sistemine yeni ikincil kullanıcı dâhil olması ya da var olan ikincil kullanıcılardan bir ya da birkaçının sistemden çıkması



Şekil 3.1 Tüm yapı durum-geçiş diyagramı

olabilir. Sistem, bu tetikleyicileri oluşturabilecek faktörleri belli zaman aralıklarında kontrol edecek bir değişiklik olduğunda işleyen duruma geçecek şekilde tasarlanmıştır. Sistem bu tetikleyicilerden birine cevap vermek üzere işleyen durumda çalıştığı esnada başka bir tetikleyici meydana gelirse onu o an için hesaba katmayacak çalışması bittiği anda eldeki değerlerle kıyaslayacak ve bir değişiklik varsa o zaman işleme başlayacaktır. Bu sayede sistem hesaplamaları yaparken birden fazla tetikleyicinin oluşması durumu tek bir seferde ele alınacak ve bu noktada bir yığılma yaşanması engellenmiş olacaktır. Ayrıca bu şekilde çalışma, sistemin sadece son bilinen duruma göre hesap yapmasını sağlayacak, arada gerçekleşen diğer durumlar için gereksiz hesaplamaları engelleyecektir.

Tez çalışmasında birincil kullanıcı durumları, maksimum kanal sayısı kadar bitlik bir girdi olarak düşünülmüştür. Bu, birincil kullanıcı durum bilgisinin bulanık olduğu durumda maksimum kanal sayısı kadar üyelik değeri anlamındadır. İkincil kullanıcılar ise ihtiyaçlarını bölüm 3.2.6 da açıklanan maliyet değerleri olarak bildirirler. Bu doğrultuda sistem, tetikleyicileri;

- P, birincil kullanıcıların durumunu
- $g_u \mid u = 1,2, \dots, U$, kullanıcı maliyetlerini
- $i_u \mid u = 1,2, \dots, U$, kullanıcı durumlarını

göstermek üzere Çizelge 3.1'de gösterilen tabloyu sürekli güncel tutarak saptayabilir.

3.2.4 Geçilebilecek durumların hesaplanması ve seçilmesi

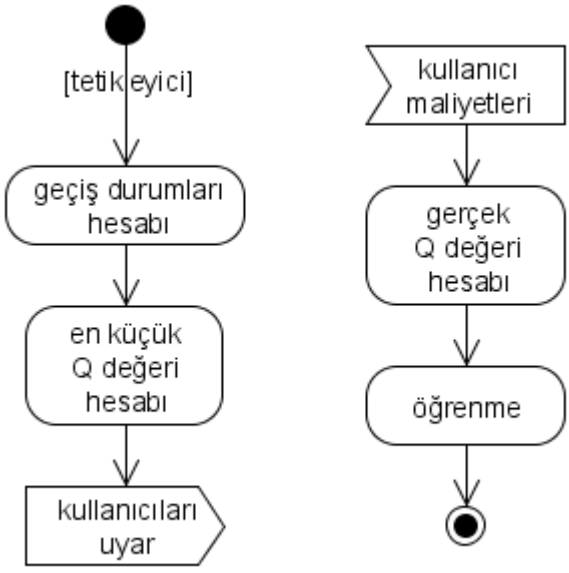
Sistem, işleyen durumunda yani hem başlangıç durumundan hemen sonra hem de herhangi bir tetikleyici olay gerçekleştikten sonra, bazı işlemler yapmak durumundadır (Şekil 3.2). Bunlardan ilki yapılabilecek hareketlerin tümünü hesaplamaktır. Bu bir nevi dinamik şekilde durum geçiş çizelgesi oluşturma işlemidir. Yukarıda bahsedilen destekleyici öğrenme kapsamındaki durumlar, BR sistemindeki ikincil kullanıcıların kullandığı kanallar şeklinde düşünülmüştür. Dolayısıyla belli bir sayıda kanalın var

Çizelge 3.1 Tetikleyici saptama tablosu

0	P	
1	i_1	g_1
2	i_2	g_2
…	…	…
U	i_U	g_U

olduğu bir ortamda belli sayıdaki kullanıcıların alabileceği farklı kanal yapıları durum uzayını oluşturmaktadır. Bu farklı kanal işgal etme durumları sadece ikincil kullanıcıların sayısına bağlı olarak değil, birincil kullanıcıların durumlarına bağlı olarak da değişmektedir. Dolayısıyla durum uzayı toplam kanal sayısına bağlı olarak ciddi miktarda büyüyebilmektedir. Bu, normalde sabit bir bilgidir fakat büyüklüğü göz önüne alındığında gerektireceği kapasite çok ciddi miktardadır. Kapasitenin büyüklüğü aynı zamanda bu bilgi içinde arama yapmayı da zorlaştıracaktır. Tüm bunlar göz önüne alındığında mümkün durum geçişlerinin o anki ikincil kullanıcı sayısı ve birincil kullanıcı durumlarına bağlı olarak hesaplanması daha mantıklı görünmektedir.

Destekleyici öğrenme disiplini içindeki yapının özel bir durumu olarak



Şekil 3.2 İşleyen durum yapısı

nitelendirilebilecek farklılık ise sistemin sadece o anki tetikleyiciye bağlı olarak hesap yapmasının gerekliliğidir. İlerleme maliyeti fonksiyonunda ilerideki kararların da önemsenmesini sağlayan γ indirim faktörü bu yapı için sıfırdır. Bunun sebebi ise sistemin bir sonraki durum geçişinin niteliği belli olmayan bir tetikleyiciye bağlı olmasıdır. Dolayısıyla bir sonraki durumlar ve geçiş ihtimalleri belirsizdir.

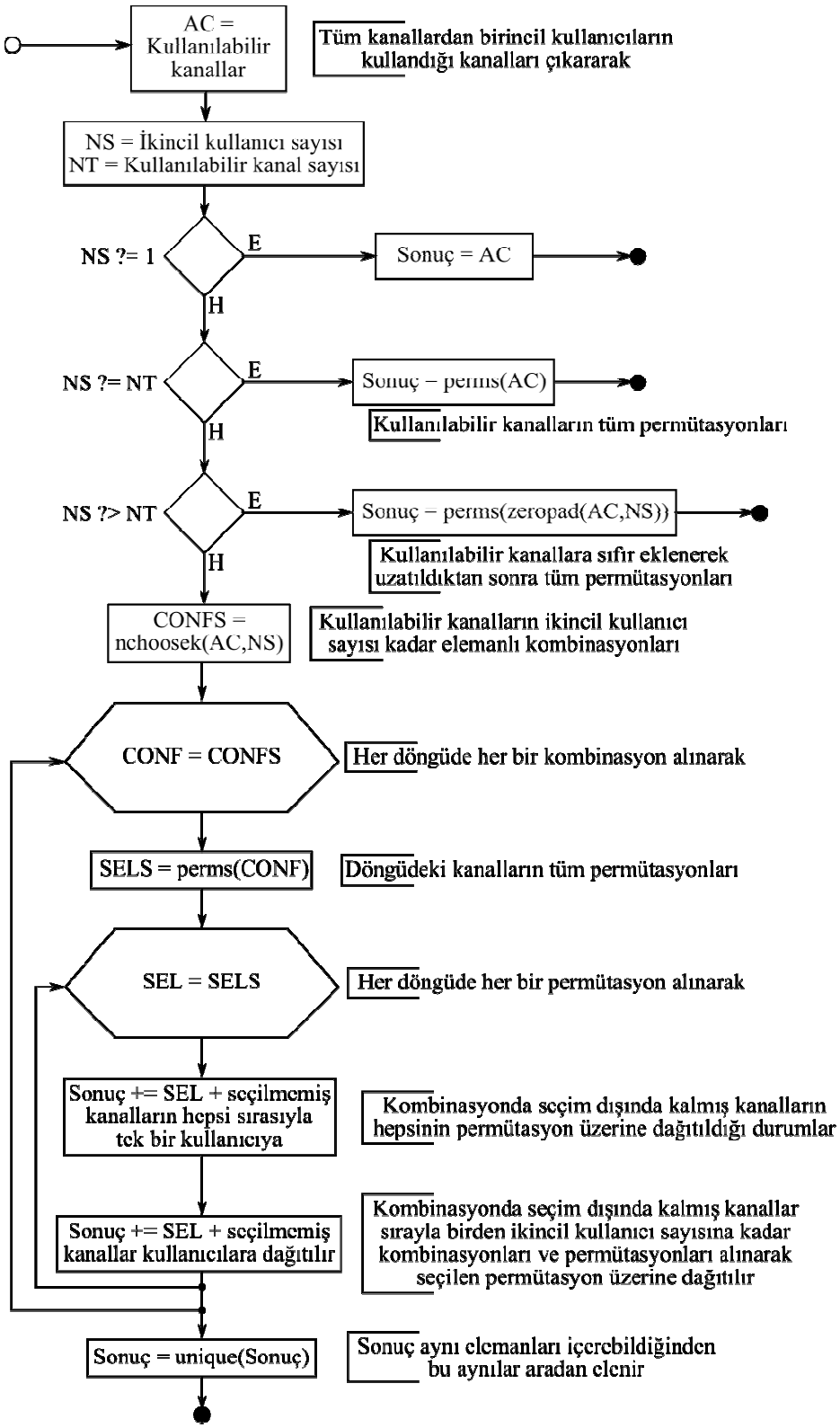
Bunlara bağlı olarak yapılabilecek geçişlerin hesaplanması için şöyle bir algoritma geliştirilmiştir: Fonksiyon parametre olarak sistemdeki tüm kanalları, birincil kullanıcıların durumunu ve ikincil kullanıcı sayısını alır. Çıktıda ise ikincil kullanıcı sayısı kadar uzunlukta olan vektörlerden oluşan bir dizi elde edilir. Dizideki her bir vektör ikincil kullanıcıların alacağı kanalları ifade eder.

1. Öncelikle birincil kullanıcıların işgal ettiği kanallar tüm kanallar arasından elenerek ikincil kullanıcılara verilebilecek kanallar bulunur.
2. Fonksiyon, sadece bir ikincil kullanıcı olduğunda tek ihtimal olan tüm kullanılabilir kanalların bu kullanıcıya verilmesi durumunu doğrudan döner.
3. İkincil kullanıcı sayısı kullanılabilir kanal sayısına eşit olduğu durumda ise fonksiyon kullanılabilir kanalların tüm permütasyonlarını döner.
4. Benzer şekilde ikincil kullanıcı sayısının kullanılabilir kanal sayısından fazla olduğu durumlarda fonksiyon, kullanılabilir kanallara ikincil kullanıcı sayısına eşit oluncaya kadar sıfır ilave eder ve oluşan yeni dizinin tüm permütasyonlarını döner. Bu sayede bazı kullanıcıların hiç kanal almadığı durumlar hesaplanmış olur.
5. İkincil kullanıcı sayısının kullanılabilir kanal sayısından az olduğu durumda kullanılabilir kanalların ikincil kullanıcı sayısı kadar eleman içerecek kombinasyonları hesaplanır.
6. Bir döngü içinde kombinasyonlardan bir eleman seçilerek o elemanın permütasyonları hesaplanır. Bunun amacı seçilen kanalların farklı kullanıcılara atanabilme ihtimalinin kapsanmasıdır.
7. Başka bir döngü içinde hesaplanan permütasyonlardan biri seçilir. Atanmamış kanal kalmaması gerektiği için kombinasyonda seçilmemiş kanallar kullanıcılara dağıtılmalıdır.

8. Öncelikle seçilmemiş tüm kanalların tek bir kullanıcıya atandığı durumlar hesaplanır ve çıktıya eklenir.
9. Daha sonra birkaç iç içe döngü vasıtasıyla seçilmemiş kanalların tüm kombinasyonları ve bu kombinasyonların tüm permütasyonları hesaplanarak kullanıcılar üzerine dağıtılır.
10. Son olarak hazırlanan çıktı içindeki aynı elemanlar elenir.

Algoritmanın akış diyagramı Şekil 3.3'te verilmiştir.

Bu algoritma görüldüğü üzere bölüm 3'de bahsedilen birincil kullanıcı bilgisinin kesin olduğu durum için geçerlidir. Bu bilginin bulanık olduğu durumda, muhtemel geçişler üyelik değerleriyle beraber değerlendirilmeli ve yapay sinir ağı, bu bilgiyi de işleyecek şekilde düzenlenmelidir.

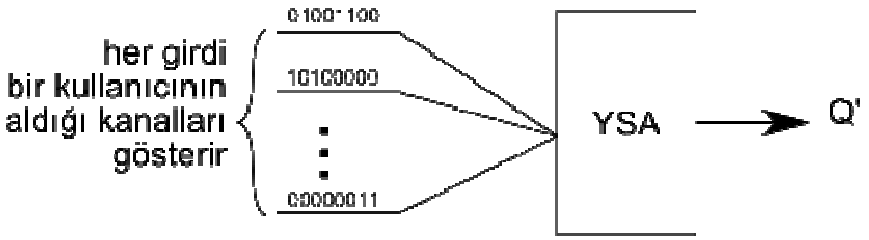


Şekil 3.3 Geçiş durumları hesaplama algoritması

3.2.5 En küçük Q değerinin hesaplanması

Geçiş yapılabilecek durumların her biri ikincil kullanıcılar için farklı maliyetler oluşturacaktır. Tasarlanan yapı bu maliyetlerden en küçüğünü oluşturacak geçişi hesaplamalı ve kullanıcıları bu geçişe yönlendirmelidir. Sistem içinde yer alan ileri-beslemeli yapay sinir ağının görevi en küçük maliyeti oluşturacak durum geçişini tahmin etmektir. Bu işlem yapay sinir ağının girişine geçiş yapılacak durumun verilmesi, çıktısında ise kullanıcı maliyetlerinin lineer bir fonksiyonu olan Q değerinin beklenmesi vasıtasıyla gerçekleştirilebilir. YSA, sürekli olarak gerçek Q değeri ile eğitilerek çıktısının doğruluğu iyileştirilebilir.

Bu çalışmada tasarlanan sistem içindeki ileri-beslemeli YSA'nın girdi sayısının sabit olması gerektiği düşünülmüştür. Diğer tarafta sistemin değişken sayıda ikincil kullanıcıyı işleyebilmesi gerektiğinden, değişken yapıdaki durum bilgisinin YSA'ya gönderilebilmesi gerekir. Tez çalışması bu problem için iki çözüm önermektedir. İlk çözüm, sistem içindeki toplam kanal sayısı sabit olduğundan bir kanal atama durumunda en fazla toplam kanal sayısı kadar ikincil kullanıcı bulunabileceği gerçeğinden hareket eder. İstekte bulunan kullanıcı sayısı toplam kanal sayısından fazla olsa da hepsini aynı durum içinde bir yere yerleştirmek mümkün olmayacaktır. Örneğin 16 kanallı bir sistemde bir durumda en fazla 16 ikincil kullanıcı atanabilir. Dolayısıyla YSA girdileri kanal atama durumu içindeki kullanıcıları temsil edecek şekilde, aynı örnekten hareketle 16 girdili, ayarlanabilir. YSA kullanıcıların aldıkları kanalların toplam kanal sayısı kadar bitlik bir sayıyla ifade edildiği bir girdiyle beslenebilir (Şekil 3.4). Ancak bu durum YSA'nın girdilerinin her birinin bir kullanıcıya sabitlendiği anlamına gelir. Zira YSA bu girdiye göre kullanıcının maliyet fonksiyonunu tahmin edecektir. Bu, iletişim isteğinde bulunan kullanıcıların zaman içerisinde değişeceği göz



Şekil 3.4 Her girdinin bir kullanıcıyı gösterdiği YSA yapısı

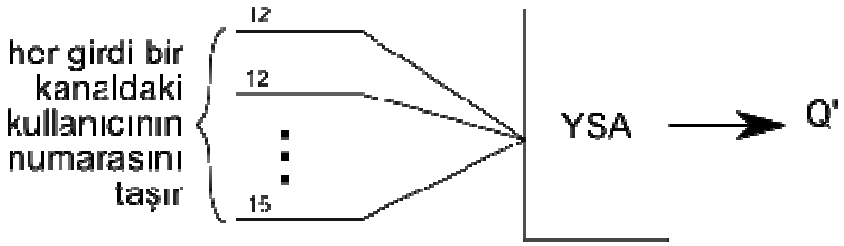
önünde bulundurulursa hatalı sonuçlara sebep olabilir. Örneğin ikinci girdisinde belli bir tip istekte bulunan A kullanıcısına göre tutarlı çıktılar üreten bir YSA, kullanıcının sistemden çıkması ve o girdinin başka bir tip istekte bulunan B kullanıcısının değerlerini gösterecek şekilde düzenlenmesi gerektiğinde yeni maliyet değerini yeterince iyi tahmin edemeyecektir. Diğer taraftan bu durum YSA'nın bir sonraki eğitiminde düzeltilebilir.

İkinci çözüm ise, YSA'nın girdilerinin BR ortamındaki her bir kanala karşılık gelecek şekilde ayarlanmasını öne sürer (Şekil 3.5). Her biri ayrı bir kanal gibi çalışacak girdiler, o kanal üzerindeki ikincil kullanıcının tanımlayıcı numarasını taşıyacak bir değer ile beslenebilirler. Birincil kullanıcılar tarafından kullanımda olan kanallara karşılık gelen girdilere ise sıfır gönderilerek üzerlerinde işlem yapılmayacağı tanımlanabilir. Bu yaklaşım ilkine göre daha dayanıklı görünse de YSA'nın daha karmaşık olmasını gerektirecektir. Zira YSA bu sefer girdileri yerine girdilerine gönderilen değerleri kullanıcılarla eşleştirmeye çalışacaktır. Bu da YSA'nın değerlere karşılık maliyet fonksiyonlarını hatırlaması gerekeceğini dolayısıyla daha fazla nörona ihtiyaç duyacağını gösterir.

Benzetimlerde iki farklı YSA yapısı da denenmiş olup performans ve değişikliklere verdikleri tepkiler açısından karşılaştırmaları yapılmıştır.

3.2.6 İkincil kullanıcıların yerleştirilmesi

Minimum Q değerini üreten kanal atama yapısındaki bilgiler, ilgili kullanıcılara kontrol kanalı üzerinden iletilir. İkincil kullanıcılar aldıkları bilgilerle gerekli kanal değişikliklerini yaparlar. Kullanıcılar, yeni atandıkları kanallardan oluşan maliyetlerini hesaplarlar. Kavramsal Q-öğrenme yapısındaki maliyet fonksiyonu, durum değiştirme



Şekil 3.5 Her girdinin bir kanalı gösterdiği YSA yapısı

hareketini de bir parametre olarak kabul etmektedir. Bu çalışmada öne sürülen yapıda ise hareketin maliyete etkisi ihmal edilmiştir. Zira sistem sadece bir tetikleyici oluştuğunda kullanıcıları kanal değiştirmeye yönlendirmektedir. Bir tetikleyici oluşmadığı sürece kullanıcılar atandıkları kanalları kullanacaktırlar ve bu kullanım süresi değişiklik yapacakları süreye kıyasla çok daha uzundur. Ayrıca atandıkları kanal yapısı iletişim kaliteleri üzerine doğrudan etkiliyken yapılması gereken değişiklik dolaylı olarak etkileyecektir.

Hesapların kullanıcılar tarafından yapılıyor olması sistemdeki maliyet fonksiyonunun kullanıcılara bağlı olduğu ve gerçek Q değerinin onlar tarafından hesaplandığıdır. Bu fonksiyon kullanıcının o anki iletişim ihtiyaçlarını karşılayabilecek sayıda kanal atanmış olmasına, kullanılan modülasyon türüne göre kanalların birbirlerine yakınlığına ya da şebeke tarafından kullanıcıya atanan öncelik kademesine göre işlem yapabilir. Hesaplanan değerler merkezi sisteme gönderilir ve gerçek Q değeri orada hesaplanır. Merkezi sistem bu hesabı direkt olarak toplamak ya da kullanıcı maliyetlerini farklı ağırlıklarla çarparak toplamak gibi çeşitli kıstaslara göre değerlendirmek suretiyle yapabilir.

Sistemin esnekliğini sağlayan noktalardan biri burasıdır. Zira kullanıcı fonksiyonlarının hesaplama mantığının ya da hesaba katılması gereken parametrelerin değişmesi fazlasıyla muhtemeldir. Örneğin yukarıda bahsedilen kanal değişikliklerin performansı ciddi miktarda etkilediği saptanırsa maliyet fonksiyonları bu parametreyi de hesaba katacak şekilde değiştirilmesi gerekebilir. Böyle bir değişiklik gerektiğinde tek yapılması gereken kullanıcıların maliyet hesapladıkları fonksiyonu basit bir şekilde güncellemektir. Benzer şekilde kullanıcı maliyet değerlerinden Q değerini hesaplayacak yapı da isteğe göre düzenlenebilir. Q değerini tahmin eden yapay sinir ağının bu gibi değişikliklere tepki vereceği ve yeni fonksiyonlara hızlıca yakınsayacağı düşünülmüştür.

Maliyetlerin merkezi sistemden bağımsız kullanıcılar tarafından hesaplanıyor olması, kullanıcıların bu fonksiyonu merkezi yapının haberi olmadan değiştirebileceği dolayısıyla maliyet çıktılarında aykırı değerler elde edilebileceği düşünülebilir. Böyle

bir etkinin var olması muhtemel şebekelerde, merkezi sistemin Q değeri hesaplama fonksiyonu bunu algılayacak ve önlem alacak şekilde hazırlanabilir. Örneğin sürekli yüksek maliyet bildiren bir kullanıcıya düşük bir katsayı uygulanarak muhtemel aykırı değerlerin etkisi bastırılabilir.

Benzetimlerde kullanıcı maliyet hesapları için beş farklı fonksiyon kullanılmıştır:

- Kullanıcı önceliği belirleyicidir

$$g_u(i_n, a_n, j_n) = \frac{p_u * N}{5c(j_n)}\ ,\ p_u \in \{1,2,3,4,5\} \tag{3.1}$$

- Bant genişliği ihtiyacı belirleyicidir

$$g_u(i_n, a_n, j_n) = \frac{c(j_n)}{R_u} \tag{3.2}$$

- Hem kullanıcı önceliği hem bant genişliği ihtiyacı belirleyicidir

$$g_u(i_n, a_n, j_n) = \frac{R_u + p_m(p_u)}{c(j_n)},\ p_u \in \{1,2,3,4,5\} \tag{3.3}$$

$$p_m(x) = \frac{x-3}{3} \tag{3.4}$$

- Kanal dağınıklığı ve bant genişliği ihtiyacı belirleyicidir

$$g_u(i_n, a_n, j_n) = \frac{R_u - s_{mu}(j_n)}{c(j_n)} \tag{3.5}$$

$$s_{mu}(x) = \frac{2s(x)}{R_u + 1} - 1 \tag{3.6}$$

- Kanal dağınıklığı, kullanıcı önceliği ve bant genişliği ihtiyacı belirleyicidir

$$g_u(i_n, a_n, j_n) = \frac{R_u + p_m(p_u) - s_{mu}(j_n)}{c(j_n)} \quad (3.7)$$

$g_u(i_n, a_n, j_n)$, u kullanıcısı için i_n durumundan a_n hareketiyle j_n durumuna geçişte oluşacak maliyeti ve N de sistemdeki toplam kanal sayısını ifade eder. Maliyet kullanıcı tarafında sadece o kullanıcıya bildirilen kanal durumuyla hesaplandığından i_n ve j_n sistemin durumunu değil sadece o kullanıcının durumunu göstermektedir. p_u, u kullanıcısının önceliğini gösterirken, R_u ise u kullanıcısının bant genişliği ihtiyacını kanal sayısı cinsinden ifade eden bir değerdir. $c(j_n)$, u kullanıcısının j_n durumunda aldığı kanalların sayısını hesaplar. $s(x)$, x durumuyla ifade edilen kanalların dağınıklığını, durum içindeki her birbirine bitişik kanal için artan bir değerle ifade eder.

Söz konusu fonksiyonlar sadece maliyet hesaplarının nasıl yapılabileceği ile ilgili örnek teşkil etmektedir. Pratik uygulamalarda farklı katsayılarla kullanılarak parametrelerin maliyet üzerine etkisi düzenlenebilir ya da fonksiyonlar düzenlenerek farklı parametrelerin hesaba katılması sağlanabilir. Maliyet fonksiyonunun detayı fazlasıyla kurulan şebekenin amacına bağlıdır dolayısıyla bu konudaki tartışma kapsam dışı tutulmuştur.

Benzetimlerde bir kullanıcının bu fonksiyonlardan herhangi biriyle hesap yaptığı durum izlenmiştir. Çalışmada kullanıcıların maliyet değerleri için;

1. Başlangıçta kullanıcıların her birine rastgele bir fonksiyon atandığı ve benzetim süresince sadece seçilen fonksiyon üzerinden hesap yapıldığı yapı,
2. Başlangıçta atanan fonksiyonun bazı zamanlarda değiştirildiği yapı,

ve gerçek Q değeri için ise;

3. Benzetim süresi boyunca sadece ortalama alan yapı,
4. Belli bir noktadan sonra kullanıcıları birbirlerinden farklı değerlendiren yapı,

ayrıca araştırılmıştır. Yukarıdaki ikinci ve dördüncü maddelerdeki yapının incelenmesi, yapay sinir ağının bu değişikliğe uyum sağlayıp sağlayamadığını görmek açısından özellikle önem teşkil etmektedir.

3.2.7 Öğrenme

Sistem içindeki öğrenilmesi gereken işlem Q değerine yakın bir değer üretmek olduğundan yapay sinir ağı sürekli olarak bunu gerçekleştirmek için eğitilmelidir. Tasarlanan yapı bu noktada kavramsal Q-öğrenme yapısından farklı değildir. Yapay sinir ağı, en başta Q değerine yakın sonuç ürettiği için seçilen kanal yapısı girdisi ve hedef Q değeri kullanılarak geri-yayınım algoritmasıyla eğitilir. 3.2.4'te bahsedilen γ indirim faktörünün sıfır olması sadece ilerleme maliyeti hesabı için geçerli olup burada yapılan Q-öğrenme için geçerli değildir. 2.4'te

$$Q_n^{hedef}(i_n, a_n) = g(i_n, a_n, j_n) + \gamma \min_b Q_n(j_n, b) \tag{3.8}$$

olarak tanımlanan hedef Q değeri içindeki γ indirim faktörü benzetimler için farklı değerler kullanılarak aranmıştır. Diğer taraftan $\min_b Q_n(j_n, b)$ işlemi j_n durumundaki gerçek maliyet bilinmediğinden yine YSA üzerinden tahmin yoluyla elde edilmiştir. Bu doğrultuda YSA'nın kendi kendini eğittiği söylenebilir.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Tez çalışmasının bu kısmı, tasarlanan sistemin farklı koşullardaki çalışmasını izleyebilmek adına yapılan benzetimleri, sonuçları ve yorumları içermektedir. Tüm benzetimler toplam 8 kanalın bulunduğu ortamlar için gerçekleştirilmiştir. Benzetimler dijital bilgisayarlar yardımıyla yapıldığından ortamda daha fazla kanalın bulunduğu sistemler için hesaplamaların süresi kabul edilebilir zamanlardan çok fazladır. Bu konuyla ilgili öneriler tezin son bölümünde yer almıştır.

Genellikle farklı başlangıç senaryoları için tekrarlanan benzetimlerde birincil kullanıcıların işgal edebileceği kanallar en fazla toplam kanal sayısının yarısı yani 4 kanal olabilecek şekilde sınırlanmıştır. Zira daha yoğun durumlar için kullanılabilecek kanal sayısı hesaplamaların fazlasıyla kolaylaştığı sayılara düşmektedir. Bu da kurulan sistemin en başta kabullendiği ortamda çok sayıda durum olması durumuyla çelişmektedir. Durumların sayısının az olduğu ortamlarda burada tasarlanan yaklaşık Q değeri öğrenmesi yapılan sistem yerine gerçek Q öğrenmesi ve hatta bir kural döngüsü algoritmasının kullanılması daha mantıklı olabilir. Tez çalışmasında bu durum kapsam dışı tutulmuştur.

Sistem performansı için YSA'nın tayin ettiği hareketin oluşturacağı maliyetin, oluşabilecek en küçük maliyetler arasından kaçıncı olduğu kullanılmıştır. Bu bağlamda sistemin sürekli olarak en küçük maliyeti yani 0'ıncı değeri seçmesi beklenmiştir.

Deneyler hazırlanan benzetim ortamında arka arkaya çalışan kodlar şeklinde gerçekleştirilmiş olsa da gösterim amaçlı bir ara yüz de mevcuttur. Bu ara yüz öncelikle hazırlanan ortamı özet bir şekilde göstermekte ve peşinden ilk hesaplamaları yapmak için beklemektedir (Şekil 4.1). Buradaki ilk sütun ikincil kullanıcı numaralarını göstermektedir. İkinci sütun kanal yapısını göstermektedir. İkinci sütundaki ilk satır birincil kullanıcıların işgal ettiği kanallar anlamındadır ve bu örnek için 00000111 şeklindedir. Bunun anlamı sekiz kanallı ortamda ilk beş kanal kullanılmamakta son üç kanal ise lisanslı kullanıcılar tarafından kullanılmaktadır. Üçüncü sütun ikincil kullanıcıların bildirdiği maliyet değerleridir ve başlangıç durumunda 1 olarak

```
    'no'          'kanal'       'maliyet'    'oncelik'    'strateji'    'gereksinim'
    'birincil'    '00000111'    '0.0000'     '0'          '0'           '0'
    '1'           '00000000'    '1.0000'     '4'          '2'           '8'
    '2'           '00000000'    '1.0000'     '1'          '2'           '8'
    '3'           '00000000'    '1.0000'     '3'          '2'           '1'
    '4'           '00000000'    '-1.0000'    '2'          '3'           '8'
    '5'           '00000000'    '-1.0000'    '4'          '1'           '8'
    '6'           '00000000'    '-1.0000'    '4'          '3'           '7'
    '7'           '00000000'    '-1.0000'    '1'          '3'           '8'
    '8'           '00000000'    '-1.0000'    '4'          '5'           '6'

Ilk hesaplamalar icin bir tusa basin
```

Şekil 4.1 Benzetim başlangıcı

atanmıştır. Dördüncü sütun ikincil kullanıcıların şebeke tarafından belirlenen öncelik değerlerini ifade eder. Bu değer deneylerde 1 ile 4 arasında bir değeri alacak şekilde rastgele seçilmektedir. 1 değeri en öncelikli 4 değeri ise en az öncelikli olmayı ifade eder. Beşinci sütun strateji olarak adlandırılır ve kullanıcıların maliyetlerini hesaplamakta kullanacakları fonksiyonu gösterir. Bu değerin sayısal olarak bir anlamı yoktur, sadece 3.2.6'da bahsedilen maliyet hesaplama fonksiyonlarından birini ifade etmektedir. Bunlar;

1. Kullanıcı önceliğinin
2. Bant genişliği ihtiyacının
3. Hem kullanıcı önceliği hem bant genişliği ihtiyacının
4. Kanal dağınıklığı ve bant genişliği ihtiyacının
5. Kanal dağınıklığı, kullanıcı önceliği ve bant genişliği ihtiyacının

belirleyici olduğu fonksiyonlardır. Son sütun ise ikincil kullanıcıların kanal gereksinimlerini gösterir. Hazırlanan ortamda sekiz kanal bulunduğundan bu değer bir ile sekiz arasında olabilir.Tuşa basıldığından benzetim ilk hesaplamalarını yapacak ve inceleme için beklemeye geçecektir (Şekil 4.2). Benzetim yaptığı işlemleri adımlar halinde gösterir. "Sistem kuruluyor" adımında benzetimin değişkenleri hazırlanır. "Geçiş durumları hesaplanıyor" adımında bahsedildiği üzere verilen durum için yapılabilecek geçiş ihtimallerinin tamamı hesaplanır. "YSA çalıştırılıyor" adımında bu hesaplanan geçiş ihtimalleri YSA'ya verilir ve tahmini maliyet değerleri elde edilir. Hemen ardından en küçük maliyeti oluşturacak hareketin detayı yazdırılır. Örnekte YSA ilk iki kullanıcıya ikişer kanal son kullanıcıya da tek kanal vermenin en küçük

```
Ilk hesaplamalar icin bir tusa basin
Sistem kuruluyor... Tamamlandi!
Gecis durumlari hesaplaniyor... 150 gecis durumu hesaplandi!
YSA calistiriliyor... Tamamlandi!
YSA ya gore en uygun hareket
    'no'          'kanal'       'kanal'
    'birincil'    '00000111'    '7'
    '1'           '00011000'    '24'
    '2'           '10100000'    '160'
    '3'           '01000000'    '64'

YSA ya gore Q degeri : 12.387217
#Gercek en uygun hareket
    'no'          'kanal'       'kanal'
    'birincil'    '00000111'    '7'
    '1'           '00011000'    '24'
    '2'           '01100000'    '96'
    '3'           '10000000'    '128'

#Maliyet : 9.000000
```

Şekil 4.2 İlk hesaplamalar

maliyeti oluşturacağını ve maliyetin de 12 gibi bir değer olacağını tahmin etmiştir. Bunun ardından sistem için aslında en küçük maliyeti oluşturacak hareket ve maliyet gösterilir. Burada dikkat edilmesi gereken şudur ki sistem ikincil kullanıcılara mümkün mertebe fazla kanal verecek şekilde çalışmaktadır. YSA sisteme dâhil olmuş kullanıcılara her zaman için en az bir kanal verecek şekilde senaryolar üretir. Bunun sebebi geçiş ihtimallerinin bu şekilde hesaplanmış olmasıdır.

Sonraki adımda sistem işlemleri gerçekleştirir ve oluşan maliyeti hesaplar (Şekil 4.3). İşlem istenirse herhangi bir değişiklik yapılmadan birkaç defa tekrarlanabilir. Sistem son olarak benzer özet yapıyla sistemin son halini gösterecektir.

Bundan sonra sistem bekleyen durumuna geçmiş olacak ve bir tetikleyici bekleyecektir. Tetikleyici için benzetime bir girdi uygulanabilir (Şekil 4.4).

```
Kullanicilarla haberlesiliyor... Tamamlandi!
Maliyet hesaplaniyor... Tamamlandi!
YSA nin verdigi durumla olusan maliyet : 9.000000
Gercek minimum maliyet : 9.000000
Islem kac defa daha tekrarlansin?
Son durum
    'no'          'kanal'       'maliyet'    'oncelik'    'strateji'    'gereksinim'
    'birincil'    '00000111'    '0.0000'     '0'          '0'           '0'
    '1'           '00011000'    '4.0000'     '4'          '2'           '8'
    '2'           '10100000'    '4.0000'     '1'          '2'           '8'
    '3'           '01000000'    '1.0000'     '3'          '2'           '1'
    '4'           '00000000'    '-1.0000'    '2'          '3'           '8'
    '5'           '00000000'    '-1.0000'    '4'          '1'           '8'
    '6'           '00000000'    '-1.0000'    '4'          '3'           '7'
    '7'           '00000000'    '-1.0000'    '1'          '3'           '8'
    '8'           '00000000'    '-1.0000'    '4'          '5'           '6'
```

Şekil 4.3 Gerçekleştirilen İşlemler

4.1 Yapay Sinir Ağının Öğrenmesi

Yapılan ilk benzetim, kullanılacak YSA'nın Q değerine yaklaştığının gösterilebilmesi amacını güder. Bu deneyde varsayılan ağırlıklarla başlatılan YSA'nın, değişmeyen durumda sahte tetikleyicilerle tekrar çalıştırılan sistem içinde eğitilerek Q değerine yaklaşması beklenmektedir. Deney, 5 defa tekrarlanmış, her tekrarda farklı bir başlangıç durumuyla, yani farklı birincil kullanıcı senaryoları ve farklı ikincil kullanıcı ihtiyaçlarıyla başlatılmış ve her birinde 10 sahte tetikleyici kullanılmıştır. Sahte tetikleyicilerden kasıt daha önce tanımlanan tetikleyicilerden herhangi birinin gerçekten oluşmaması, sistemin aynı durum üzerinde tekrardan çalıştırılması anlamındadır. Bu durumda YSA'nın problemin en genel hali için bir çözüm teşkil edip edemeyeceği irdelenmektedir.

Sonuçların değerlendirilmesinde sistemin ürettiği maliyet değerlerinin tüm oluşabilecek maliyet değerleri arasındaki yeri kullanılmıştır. Özetle sistem belli bir durum geçişi için sınırlı aralıkta maliyet değerleri üretebilmektedir. Performans kıstası ise sistemin verdiği

```
Yapilacak islem ?
0:Bitir
1:Degisiklik yapmadan tekrarla
2:Birincil kullanici degistir
3:Ikincil kullanicilardan birini degistir
4:Ikincil kullanici ekle/cikar
```

Şekil 4.4 Bekleyen durum

durum geçişi ile oluşan maliyet değerinin bu maliyet ihtimalleri arasındaki yerini ifade eder. Gerçek maliyet ihtimalleri ise tüm durum geçişleri uygulanarak hesaplanmıştır.

Deneylerde genellikle gözlenen durum sistemin beklendiği şekilde hatalı başlayıp tetikleyiciler ilerledikçe bir değere yakınsaması olmuştur (Ek 1). Ancak sistem yapılan 5 deneyin 2'sinde istenen davranış dışında optimal olmayan bir değere yakınsamıştır. Çizelge 4.1, benzetimlerin hangi değerlere yakınsadığını göstermektedir. Bu ve diğer grafiklerdeki değerler, seçilen geçiş durumunun ürettiği maliyetin, tüm geçiş durumlarının oluşturacağı maliyetler arasındaki yerini göstermektedir. %0, en düşük maliyetin, %100 ise en yüksek maliyetin oluştuğunu ifade eder. Benzetimde optimal değerin bulunamadığı 2. ve 5. deneylerde yaklaşık Q-öğrenme yapısında öngörüldüğü üzere sistemin ıraksadığı gözlenmiştir.

Benzetimlerin gerçekleştirilmesinde birçok YSA yapısı ve birçok γ değeri denenmiştir. Verilen deney sonuçları ise YSA'nın 2 gizli katmanında sırayla 30 ve 40 nöron bulunduğu γ değerinin ise 0.5 olduğu durumda elde edilmiştir. Denemeler esnasında γ'nın çok küçük değerleri için sistem hemen bir değere oturmakta ve çoğu zaman bu değer optimal olmamaktadır. Yüksek γ değerleri için ise sistem optimal bir değer bulsa da çok üzerinde durmamakta ve farklı geçiş senaryoları denemektedir. Bu durum ise en iyi geçiş ihtimalinin bulunmasını zorlaştırmaktadır. Bu farklılıklar sistemin açınsama ve kullanma stratejileri arasında dengelenmesi gerektiğinin bir ispatıdır.

YSA'nın daha az nöron içerdiği durumlarda sistem optimal değeri bulsa da sonraki denemelerinde kolayca kaybedebilmektedir. Bu problem YSA'nın öğrenmek için yeterince kapasiteye sahip olmadığı şeklinde yorumlanmıştır. Daha fazla nöron içeren

Çizelge 4.1 Yapay sinir ağı öğrenmesi benzetimi performansı

Deney	Hata
1	%0
2	%40
3	%0
4	%65
5	%0

YSA'lar ise çalışma süresini uzatmakta ama performansı etkilememektedir.

4.2 Birincil Kullanıcı Değişiklikleri

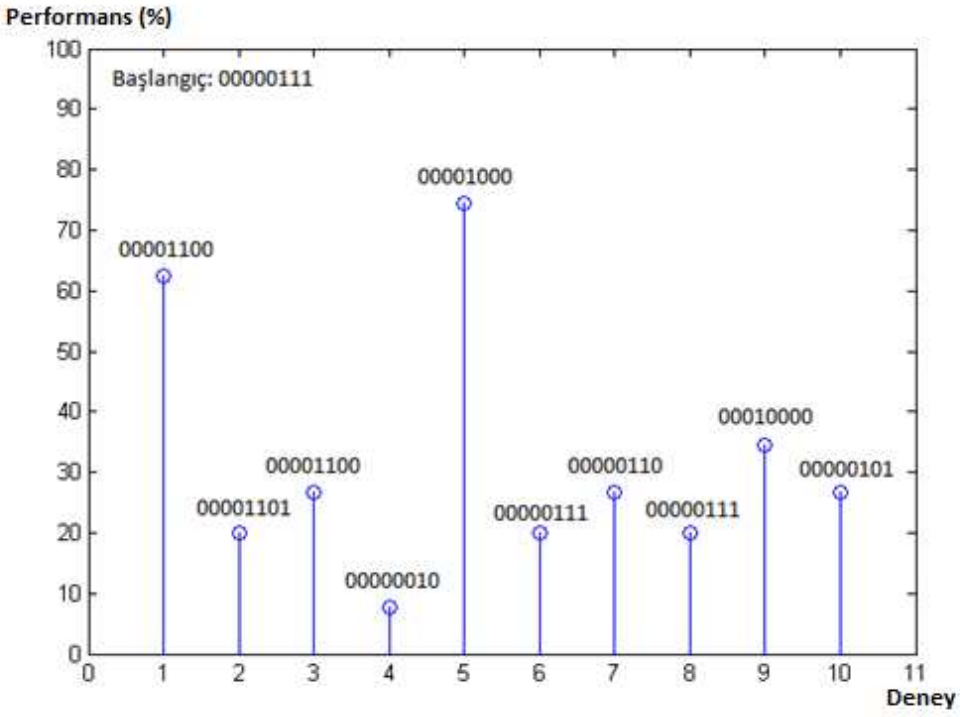
Bu benzetim, tasarlanan sistemin muhtemel birincil kullanıcı değişikliklerine uyum sağladığının gösterilmesi için gerçekleştirilmiştir. Birincil kullanıcı değişikliklerinden kasıt deney içinde tetikleyici olarak, ikincil kullanıcılar tarafından kullanılamayacak kanalların değiştirildiği ikincil kullanıcıların herhangi bir değişikliğe uğramadığı anlamındadır. Bu, sistemin bir önceki çalışmasına göre;

- yeni bir ya da birden fazla birincil kullanıcının eklenmesi,
- var olan bir ya da birden fazla birincil kullanıcının çıkması,
- ikincil kullanıcılar tarafından kullanılamayacak kanalların değiştirilmesi

olabilir. Benzetimde kullanılan YSA, başlangıç durumundaki senaryo için elle hazırlanan gerçek maliyet değerlerine göre bir kere eğitilir ve YSA'nın daha sonraki 10 tetikleyiciye uyum sağlaması beklenir. Farklı başlangıç senaryolarıyla 5 deney gerçekleştirilmiştir. Sistemin performansı 4.1'de olduğu gibi oluşan maliyetin oluşabilecek maliyet ihtimalleri arasındaki yeri ile ölçülmüştür.

Deneyler sistemin genellikle en düşük maliyeti seçemediği ama nispeten düşük maliyetleri seçtiğini göstermiştir (Ek 2). Bunun sebebi YSA'nın aslında sadece ilk durum için eğitilmiş olması ve diğer durumlar için bir bilgisinin olmamasıdır. Daha önce de belirtildiği üzere sistem sadece rastgele seçilen başlangıç durumu için bir kere eğitilmekte ve sonra oluşabilecek durumlar için herhangi bir ek düzenleme yapılmamaktadır.

Ayrıca YSA birincil kullanıcı durum bilgisini direkt olarak almamakta onun yerine ikincil kullanıcıların yerine bakarak anlamaya çalışmaktadır. Örneğin 2. deneyde YSA sadece ilk durum olan (0'lar boş 1'ler dolu kanalları göstermek üzere) 00000111 durumu için sadece ikincil kullanıcıların yerleri üzerinden bilgi sahibidir (Şekil 4.5).



Şekil 4.5 Birincil Kullanıcı Değişikliği Deney 2

Bunun anlamı YSA sadece ilk beş girdisinden bilgi geldiğinden ötürü diğer üç kanalın dolu olduğunu düşünmektedir. Peşinden gelen diğer birincil kullanıcı durumlarını ve geçiş olasılıklarını bilmemekte, sadece tahminler yürütmektedir. Daha farklı bir deyişle sistem, ikincil kullanıcıların ilk durumda boş olan ilk beş kanalda yer alması durumunu bilmekte ancak ilk durumda dolu olan ve dolayısıyla orada hiç bulunulmamış son üç kanalda yer alması durumunu bilmemektedir. Bu da YSA'nın ilgili girdilerinde daha sonradan kötü etkileyebilecek bazı ağırlıkların oluşmuş olma ihtimalini beraberinde getirmektedir.

Şekilden de görüldüğü üzere sistemin en kötü değer verdiği durum, ilk durumun tam tersi olan (son üç kanalın boş olduğu) 5. tetikleyicidedir. Buna rağmen sistemin çok kötü sonuçlar üretmediği söylenebilir. Ayrıca YSA açınsama yaparak sürekli yeni bilgiler öğrenmeye çalışacak, dolayısıyla her zaman en iyi değeri vermeyecektir. Ancak bu doğrudan gözlenebilir bir kavram olmadığından her bir tetikleyici için sonucun neden iyi olmadığını kesin olarak ifade etmek çok doğru olmayacaktır.

4.3 İkincil Kullanıcı Gereksinim Değişiklikleri

Bu benzetimde sistemdeki ikincil kullanıcıların ihtiyaçlarının değişmesi durumunda sistemin tepkisinin gözlenmesi amaçlanmıştır. Başlangıç durumunda verilen birincil kullanıcı yapısı ve ikincil kullanıcı sayısı sabit tutulmuş sadece rastgele seçilen bir ikincil kullanıcının kanal gereksinimi değiştirilmiştir. Bunun anlamı deney içindeki tetikleyiciler sadece ikincil kullanıcıların kanal gereksinimlerinin değişmesi durumunu kapsamaktadır. YSA, 4.2'de olduğu gibi sadece başlangıç senaryosu için gerçek maliyetlere göre bir kere eğitilir ve tetikleyiciler ondan sonra gönderilir. 5 farklı başlangıç senaryosuyla tekrarlanan deneyin her tekrarında 10 tetikleyici gösterilmiştir.

Bu benzetim 4.2'deki birincil kullanıcı değişikliklerine göre çok daha iyi sonuçlar üretmiştir (Ek 3). Zira daha önce de belirtildiği üzere YSA ikincil kullanıcı durumlarını doğrudan girdi olarak almaktadır. Dolayısıyla sistem, ikincil kullanıcıların ilk durumda işgal ettiği kanallar ve bu sayede oluşan maliyetler hakkında bilgi sahibidir ve bu bilgi o kanalların ifade edildiği girdilerdeki ağırlıklarda tutulmaktadır. Bu benzetim temelde YSA'nın girdi değerlerinin değiştirilmesi şeklinde gerçekleştiğinden daha iyi sonuçlar elde edilmesi çok doğaldır.

Bu deney sonuçlarında açıklanması gereken iki durumdan ilki, ilk deneydeki 9. tetikleyici için maliyet değerinin %100'den fazla görünüyor olmasıdır. Bunun sebebi dijital bilgisayarların rasyonel sayılarla yaptığı işlemlerin kesinlik arz etmemesidir; dolayısıyla benzetimin sonuçlarını etkileyecek bir önemi yoktur. Diğer konu ise 3. deneyde sistemin 10 tetikleyicide de sürekli optimal değeri bulmuş olmasıdır. Bunun sebebi ise o deneyde ikincil kullanıcıların maliyet fonksiyonlarının kanal gereksinimlerine bağlı olmamasıdır. Dolayısıyla kanal gereksinimlerini değiştirmek maliyet üzerinde etkili olmayacak, YSA'nın tetikleyicilerden önce öğrendiği yapı yeterli olacaktır. Bu deney ilgili benzetim için önemli değilmiş ve çıkarılması gerekiyormuş gibi görünse de burada durmasının sebebi YSA'nın açınsama özelliğinin sistemin kararlılığını bozacak seviyede olmadığının gösterilmesidir. Eğer YSA gerektiğinden fazla açınsama yapıyor olsaydı, daha önce öğrendiklerini önemsemeyip gereksiz ve bozucu değişikliklere teşebbüs edebilirdi. Burada ise durum bu davranışın aksine

kullanma yapısını gerçekleştirmiş ve maliyet her zaman optimal olacak şekilde kalmıştır.

4.4 İkincil Kullanıcı Eklenmesi ve Çıkarılması

Bu benzetimde kurulan sistemin ortama muhtemel ikincil kullanıcı eklenmesi ya da var olan ikincil kullanıcılardan bir ya da bir kaçının sistemden çıkması durumundaki tepkisi izlenmeye çalışılmıştır. YSA, başlangıç durumundaki gerçek maliyetlere göre bir kere eğitilmiş ve tetikleyicilere göre uyum sağlayıp sağlayamayacağı gözlenmeye çalışılmıştır. Tetikleyiciler, ortama yeni gereksinim ve önceliklerle eklenen bir ikincil kullanıcı ya da var olan bir ikincil kullanıcının ortamdan ayrılması olabilir. Deney 5 farklı başlangıç durumuyla tekrarlanmış ve her tekrarda 10 tetikleyici kullanılmıştır.

Deneyler göstermiştir ki, sistem tüm senaryolar içinden en kötü performansı burada göstermiştir (Ek 4). Tetikleyicilerin neredeyse yarısında en yüksek maliyeti üretecek geçiş durumunu seçen YSA'nın bu şekilde davranması şöyle açıklanabilir. YSA'nın ilk eğitimi sadece üç adet ikincil kullanıcı için gerçekleştirilmiş, YSA'nın diğer ikincil kullanıcılarla ilgili girdileri boş bırakılmıştır. Dolayısıyla YSA bu girdilerle ilgili hiçbir bilgiye sahip değildir. Bu problem tabii ki sadece yapıya yeni kullanıcı eklendiğinde meydana gelecek performans kaybını açıklamaktadır. Diğer taraftan sistemden bir ikincil kullanıcı ayrılması durumda ise daha önce aktif olan bir girdi çalışmamaya başlayacaktır. Bu durumda da o girdiye bağlı sinaptik ağırlıklar etkisini yitireceğinden YSA öğrendiklerini unutmuş gibi olacaktır. Bu da performansı o ağırlıklarda yüklü olan bilginin miktarına bağlı olarak etkileyecektir.

Bu deney sonuçlarında açıklanması gereken bir nokta ise bazı tetikleyicilerde oluşan maliyetin şekillerde görünmemesidir. Bunun sebebi sistemin o tetikleyicilerde tek bir maliyetin oluşabileceği duruma girmiş olmasıdır. Bu boş kanal sayısı ikincil kullanıcı sayısına eşit olduğu durumda gerçekleşebilir. Zira bu durumda yapılması gereken tüm kullanıcılara birer kanal atanmasıdır. Diğer olasılık ise sistemde tek bir ikincil kullanıcı olmasıdır ki bu durumda yapılacak hareket normal olarak tüm boş kanalların bu

kullanıcıya atanmasıdır. Bunlar gibi tek bir maliyet ihtimali olan bir sistemin performansının gösterilmesi çok anlamlı değildir.

5. SONUÇ

Tasarlanan sistem teorik arka planda açıklanan hemen her kavram ile tutarlı bir şekilde çalışmıştır. Destekleyici öğrenmeden gelen açınsama ve kullanma kavramları ile yaklaşık Q-öğrenme algoritmasından gelen ıraksayabilme ihtimali bilinmezlerine rağmen sonuçların tatmin edici olduğu söylenebilir. Hatta sistemin, sabit sayıda ikincil kullanıcının bulunduğu ortamlarda kullanılabilir olduğu dahi iddia edilebilir. İkincil kullanıcıların iletişim ortamına sonradan eklenebildiği ya da ortamdan çıkabildiği şebekeler için ise sistem parçaları, özellikle YSA yapısı, bazı düzenlemelerden geçirilebilir ve bu yapıya uygun hale getirilebilir. Bu konu ileride yapılabilecek çalışmalar kapsamındadır.

Sistemin geçiş olasılıklarını hesaplayan parçası tasarlanan tüm permütasyonları ve onların kombinasyonlarını hesapladığı hali daha önce de belirtildiği üzere 8'den fazla kanalın mevcut olduğu şebekeler için uygun değildir. Bu uygunsuzluk özellikle yapılan benzetimlerin fazlasıyla uzun sürmesi şeklinde kendini belli etse ve gerçek işlemcilerle uygulandığında daha hızlı olması beklense de 10'dan fazla kanal için yapılması gereken hesaplama sayısı astronomiktir. Dolayısıyla geçiş durumlarının tamamını hesaplamak hiçbir durum için uygun değildir. Bu parça, tüm geçiş durumlarının bir alt kümesini hesaplayacak farklı bir algoritmayla değiştirilmelidir. Bu algoritma muhtemel düşük maliyetli geçiş durumlarını toplayabilir ya da benzer geçiş durumlarını eleyecek bir yapıya sahip olabilir. Ancak bu konu da ileride yapılabilecek çalışmalar kapsamındadır.

Sonuç olarak çalışan bazı parçalarda değişiklikler öngörülse de tasarlanan sistem yapı olarak amacına ulaşmıştır. Bu bağlamda tez çalışması dinamik spektrum yönetimi problemi için kullanılabilecek bir sistemi destekleyici öğrenme kavramlarıyla ortaya koymuş ve bu yapıyı uygulayan bir 'ilk örnek' çalıştırmıştır.

KAYNAKLAR

Akyildiz, I.F. 2006. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. Computer Networks, Vol. 50; pp. 2127-2159.

Akyildiz, I.F., Lee, W.Y. and Chowdhury, K. R. 2009. CRAHNs: Cognitive radio ad hoc networks. Ad Hoc Networks.

Anonymous, 2008. Web sitesi. http://gnuradio.org/trac,. Erişim Tarihi: 27.10.2008.

Arbib, M.A. 2003. The Handbook of Brain Theory and Neural Networks, MIT Press, Wisconsin.

Barto, A.G. 2003. Reinforcement Learning. The Handbook of Brain Theory and Neural Networks, MIT Press, Wisconsin.

Bertsekas, D.P. and Tsitsiklis, J.N. 1996. Neuro-Dynamic Programming, Athena Scientific, Belmont.

Cabric, D., Mishra S.M. and Brodersen, R.W. 2004. Implementation Issues in Spectrum Sensing for Cognitive Radios. Proceedings of the IEEE Asilomar Conference on Signals, Systems and Computers 2004, Monterey.

Du, K.L. and Swamy, M.N.S. 2006. Neural Networks in a Softcomputing Framework, Springer.

Gandetto, M. and Regazzoni, C. 2007. Spectrum Sensing: A Distributed Approach for Cognitive Terminals. IEEE Journal on Selected Areas in Communication, Vol. 25; pp. 546-557.

Haykin, S. 1999. Neural Networks A Comprehensive Foundation, Prentice Hall, New Jersey.

Haykin, S. 2005. Cognitive Radio: Brain-Empowered Wireless Communications. IEEE Journal On Selected Areas In Communications, pp. 201-220.

Jondral, F.K. 2007. Cognitive Radio: A Communications Engineering View. IEEE Wireless Communications, pp. 28-33.

Jondral, F.K. and Blaschke, V. 2007. Evolution of Digital Radios: From Analog to Cognitive Features. Cognitive Wireless Networks: Concepts, Methodologies and Visions Inspiring the Age of Enlightenment of Wireless Communications, Springer, Dordrecht.

Kushwaha, H. 2008. Reliable Multimedia Transmission Over Cognitive Radio Networks Using Fountain Codes. Proceedings of the IEEE, Vol. 96; pp. 155-165.

Le, B., Rondeau, T.W. and Bostian, C.W. 2007. Cognitive radio realities. Wireless Communications and Mobile Computing, Vol. 7; pp. 1037-1048.

Letaief, K.B. and Zhang, W. 2007. Cooperative Spectrum Sensing. Cognitive Wireless Communication Networks, Springer, New York.

Marshall, P. 2006. Spectrum Awareness. Cognitive Radio Technology, Elsevier, Burlington.

Mitola, J. and Maguire, G.Q. 1999. Cognitive radio: making software radios more personal. IEEE Personal Communications, Vol. 6(4); pp. 13-18.

Mitola, J. 2006. Cognitive Radio Architecture : The Engineering Foundations of Radio XML, John Wiley & Sons, Inc., Hoboken.

Mitola, J. 2000. Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering, John Wiley & Sons, Inc., Hoboken.

Pawelczak, P. and Prasad, V. 2006. IEEE P19000 Working Group, IEEE Standards Association, Web sitesi. http://grouper.ieee.org/groups/scc41/. Erişim Tarihi: 09.06.2009.

Petrova, M. and Mahonen, P. 2007. Cognitive Resource Manager. Cognitive Wireless Networks, Springer, Dordrecht.

Sahai, A., Hoven, N. and Tandra, R. 2004. Some fundamental limits on cognitive radio. Proc. Allerton Conf. on Communications, Control, and Computing, Monticello.

Sivanesan, K. and Mazzarese, D. 2006. Cooperative Techiques in the IEEE 802 Wireless Standards: Opputunities and Challenges. Cooperation in Wireless Networks: Principles and Applications, Springer, Dordrecht.

Uchiyama, H. 2008. Study on Soft Decision Based Cooperative Sensing for Cognitive Radio Networks. IECE Trans. Commun., Vols. E91-B; pp. 95-101.

Watkins, C.J.C.H and Dayan, P. 1992. Q-Learning, Kluwer Academic Publishers, Vol. 8; 279-292, Boston.

EKLER

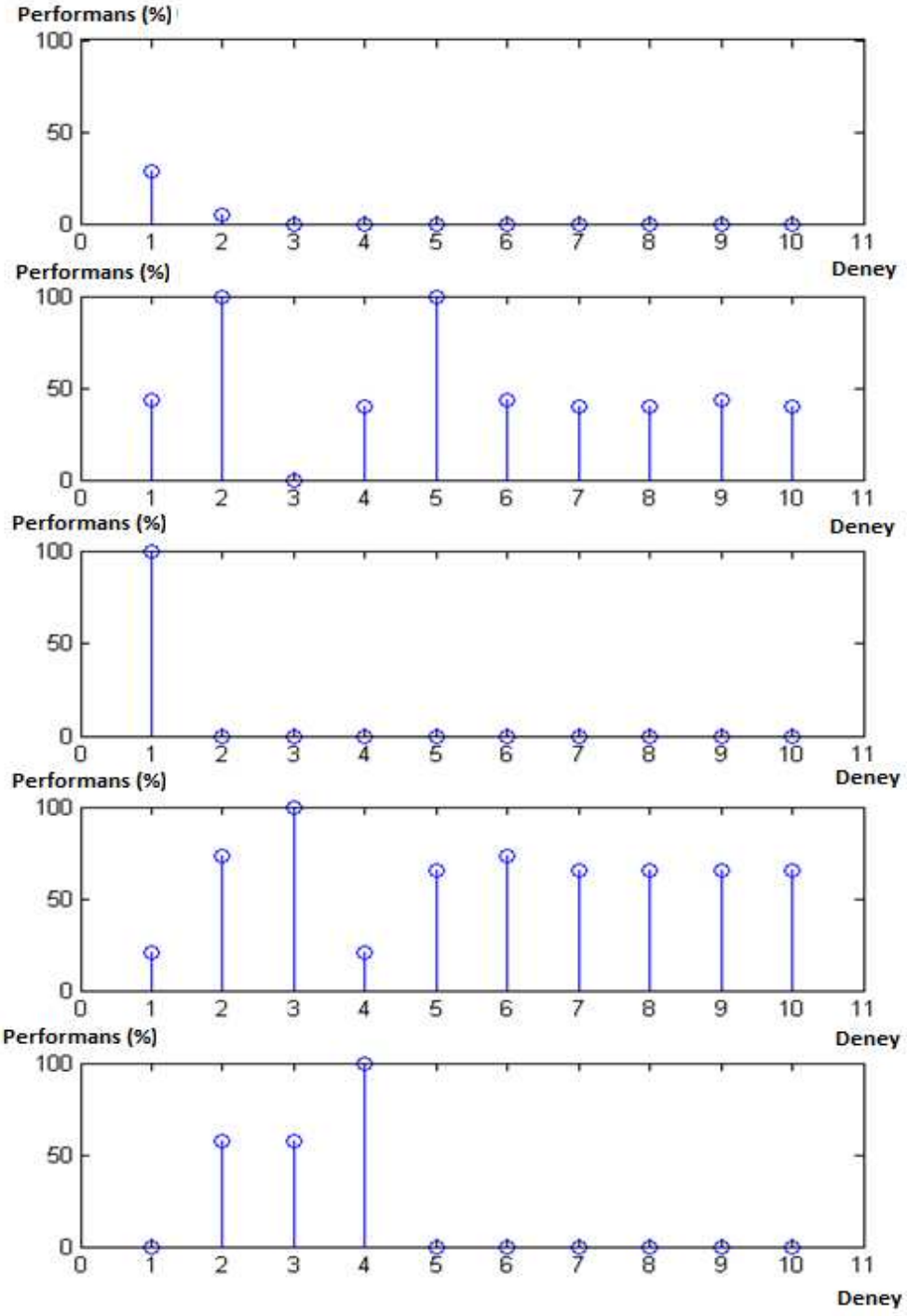
EK 1 YAPAY SİNİR AĞININ ÖĞRENMESİ BENZETİM SONUÇLARI

EK 2 BİRİNCİL KULLANICI DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI

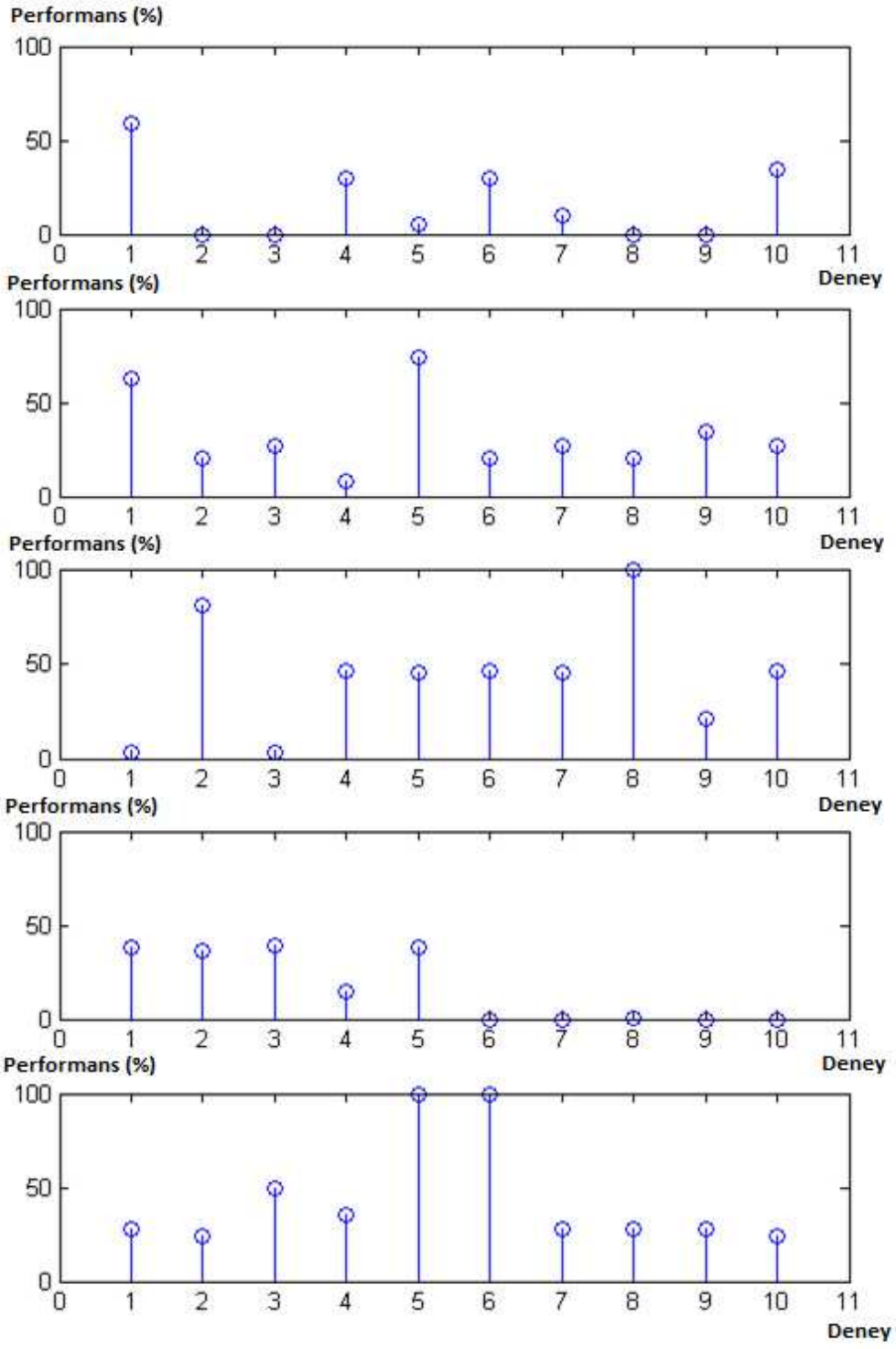
EK 3 İKİNCİL KULLANICI GEREKSİNİM DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI

EK 4 İKİNCİL KULLANICI EKLENMESİ YA DA ÇIKARILMASI BENZETİM SONUÇLARI

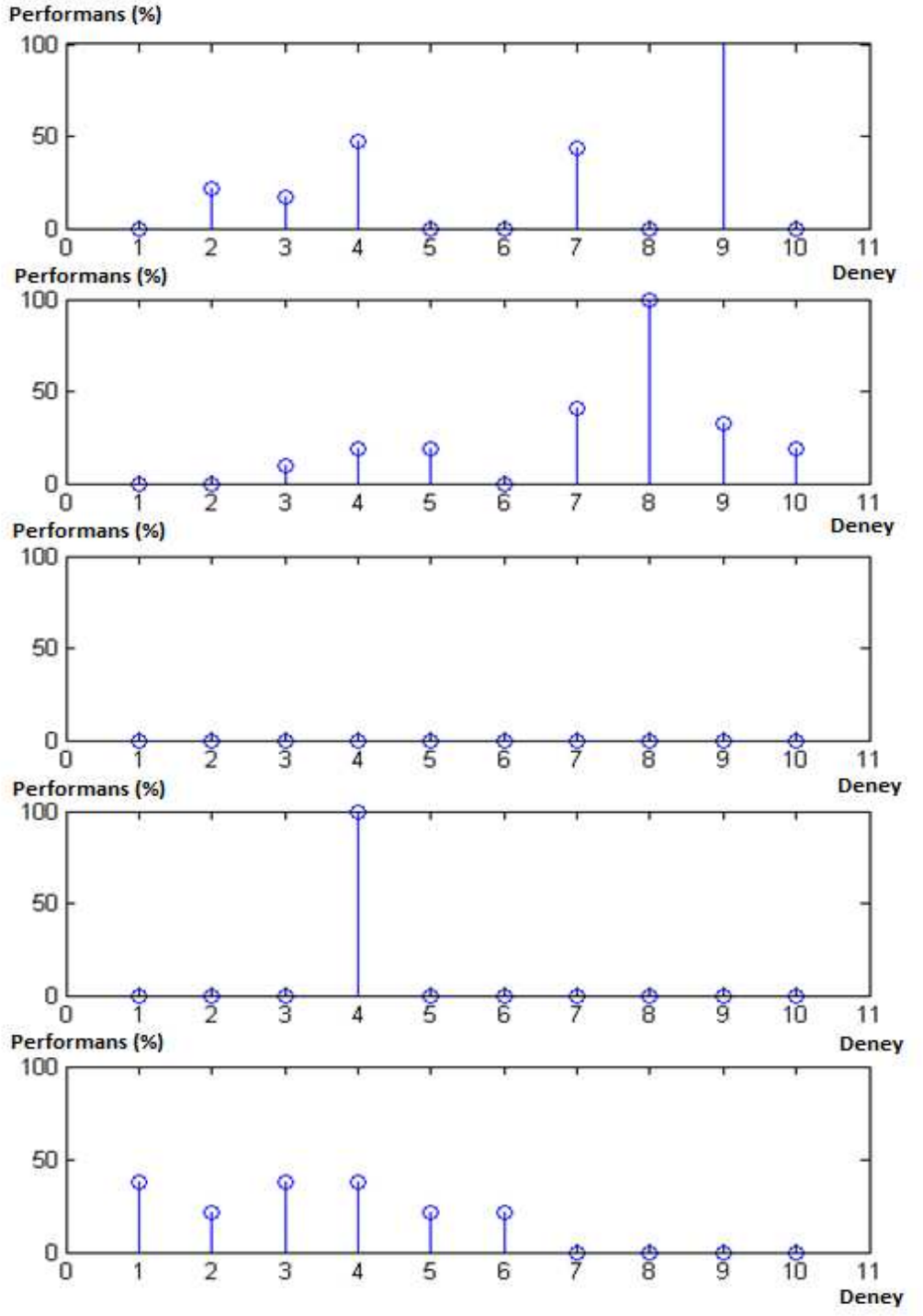
EK 1 YAPAY SİNİR AĞININ ÖĞRENMESİ BENZETİM SONUÇLARI



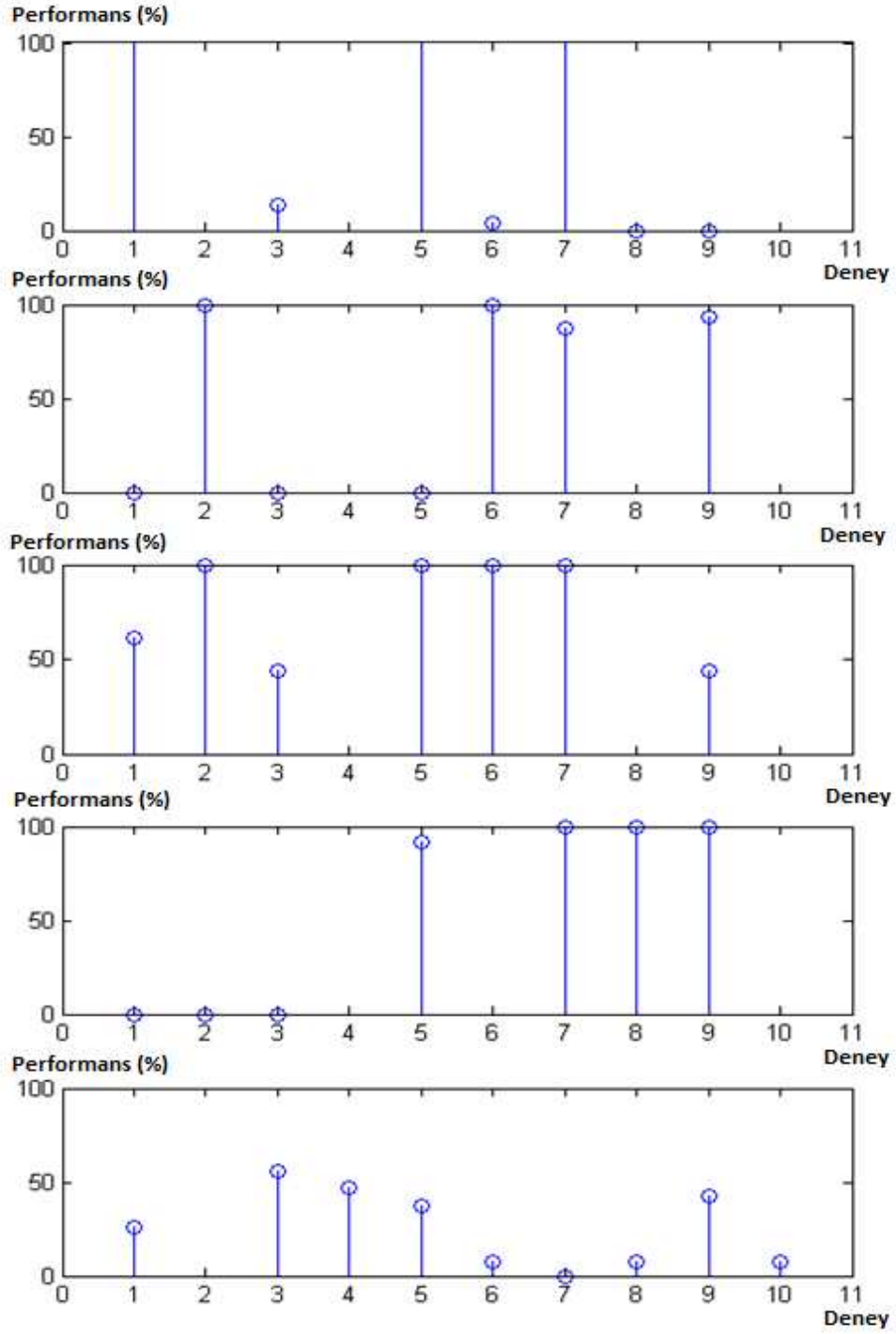
EK 2 BİRİNCİL KULLANICI DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI



EK 3 İKİNCİL KULLANICI GEREKSİNİM DEĞİŞİKLİKLERİ BENZETİM SONUÇLARI



EK 4 İKİNCİL KULLANICI EKLENMESİ YA DA ÇIKARILMASI BENZETİM SONUÇLARI



ÖZGEÇMİŞ

Adı Soyadı : Erdem AĞAOĞLU
Doğum Yeri : Bandırma
Doğum Tarihi : 21.03.1984
Medeni Hali : Bekar
Yabancı Dili : İngilizce

Eğitim Durumu :

Lise : Bandırma Şehit Mehmet Gönenç Lisesi (2001)

Lisans : Ankara Üniversitesi Mühendislik Fakültesi Elektronik Mühendisliği Bölümü (2006)

Yüksek Lisans : Ankara Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı (Eylül 2006 – Mart 2010)

Çalıştığı Kurumlar :

Ankara Üniversitesi Bilgi İşlem Merkezi, 2004-2006
Frame Bilgisayar ve Mühendislik Tic. Ltd. Şti., 2006-2008
SPP42 Yazılım Geliştirme ve Danışmanlık Ltd. Şti., 2008-

Yayınları :

Ağaoğlu, E. ve Sazlı, M. H. 2008. Bilişsel Ağlar, Habtekus 2008, İstanbul.