T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOKLU ETMEN MİMARİSİ VE TAKVİYELİ ÖĞRENME

Mehmet HACIBEYOĞLU

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

ANABİLİM DALI

Konya - 2006

ÖZET

Yüksek Lisans Tezi

ÇOKLU ETMEN MİMARİSİ VE TAKVİYELİ ÖĞRENME

Mehmet HACIBEYOĞLU

Selçuk Üniversitesi Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Ahmet ARSLAN

2006, 70 Sayfa

Takviyeli öğrenme yöntemleri, bağımsız etmenli sistemlere ve dinamik ortamlara uygulanabilir olduğundan son zamanlarda oldukça ilgi çekmiştir. Takviyeli öğrenme metotları içerisinde en popüler algoritmalardan biri Q öğrenmedir. Q öğrenme, labirent ve arama problemlerine uygulandığında optimal politikayı belirleme hızı bakımından ilgi çekici bir öğrenme metodu olarak bilinmektedir. Bunun yanı sıra durum uzayının genişlemesi Q öğrenme algoritmasında yavaşlamalara neden olmaktadır.

Bu tez çalışmamda bu problemin çözümü için yeni bir yöntem sunulacaktır. Bu yöntemde takviyeli öğrenme ile yapay sinir ağları birlikte kullanılmaktadır. Geliştirilen eylem seçme yöntemi ile etmenin q değerleri yanı sıra yapay sinir ağı çıkışlarından da faydalanması sağlanmaktadır. Böylece etmen hedefine ulaşabileceği konumlara daha hızlı bir şekilde gelebilmektedir. Deneysel sonuçlar durum uzayında gerçekleştirilen uygulamanın geçerliliğini göstermektedir.

Anahtar Kelimeler: Takviyeli öğrenme, Q öğrenme, yapay sinir ağları, labirent ve arama problemleri

ABSTRACT

Master Thesis

MULTI AGENT ARCHITECTURE AND REINFORCEMENT LEARNING

Mehmet HACIBEYOĞLU

Selçuk University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Supervisor: Prof. Dr. Ahmet ARSLAN

2006, 70 Page

The application of reinforcement learning to autonomous agent systems and dynamic states has attracted recent attention. One of the most popular reinforcement techniques is Q learning. It has been proven to produce an optimal policy under maze and searching problems. Increases the dimension of state space bring disadvantages to this algorithm.

This theses presents a novel approach to overcome this problem. The approach using together reinforcement learning and artificial neural networks. Q values and artificial neural network outputs are using together with developed action select method. Thus agent can reach the goal state rapidly. Experimental results handled on pursuit domain show the effectiveness and applicability of the proposed approach.

Keywords: Reinforcement learning, q learning, artificial neural networks, maze and searching problems

TEŞEKKÜR

Bu tez çalışmamın gerçekleşme sürecinde yardımlarından dolayı tez danışmanı hocam Sayın Prof. Dr. Ahmet ARSLAN' a teşekkür ederim.

Tez çalışmam boyunca destek ve güvenleriyle her zaman yanımda olan eşime ve aileme çok teşekkür ederim.

ÖZET i
ABSTRACT ii
TEŞEKKÜR iii
1. GİRİŞ 1
1.1. Problemin Tanımı 1
1.2. Amaç 2
2. literatür araştırması 3
3. etmen ve takviyeli öğrenme 5
3.1. Giriş 5
3.2. Öğrenme Metotları Arasındaki Farklar 5
3.3. Etmen Ne Demektir? 7
3.4. Etmenin Özellikleri 8
3.4.1. Zeka 8
3.4.2. Özerklik 8
3.4.3. Öğrenebilme yeteneği 8
3.4.4. İş birliği, ortak çalışma 9
3.4.5. Etmenin diğer özellikleri 9
3.5. Etmen Sınıflandırması 9
3.5.1. Tepkisel etmenler 10
3.5.1.1. Hedefe yönelik etmenler 10
3.5.1.2. Faydaya yönelik etmenler 11
3.5.2. İşbirlikçi etmenler 11
3.5.3. Arabirim etmenler 11
3.5.4. Mobil etmenler 12
3.5.5. Bilgilendirme etmenleri 13
3.5.6. Melez etmenler 15
3.6. Etmen ve Çevre 15
3.7. Çevresiyle Etkileşen Etmen 17
3.8. Takviyeli Öğrenme 18
3.9. Takviyeli Öğrenme Algoritması 22
3.9.1. Amaç ve ödül 22
3.9.2. Değer fonksiyonu 23
3.9.3. Değer fonksiyonunun hesaplanması 23
3.9.4. Takviyeli öğrenmenin amacı 24
3.10. Q Öğrenme 25
3.10.1. Q Öğrenme Algoritması 28
3.10.2. Açıklayıcı bir örnek 28
3.11. Takviyeli Öğrenmenin Geçmişi ve Yapılan Çalışmalara Bir Bakış 30
4. Yapay Sinir Ağlarının Takviyeli öğrenmeyle birlikte kullanılması 35
4.1. Kavram ve Terimler 35
4.2. Yapay Sinir Ağlarının Sınıflandırılması 40
4.2.1. Tek katmanlı ileri beslemeli ağlar 40
4.2.2. Çok katmanlı ileri beslemeli ağlar 40
4.2.3. Geri dönümlü ağlar 41
4.2.4. Kafes yapılı ağlar 42

4.3. Aktivasyon Fonksiyonları 42
4.4. Yapay Sinir Ağlarında Öğrenme 45
4.4.1. Güdümlü Öğrenme 45
4.4.2. Güdümsüz Öğrenme 45
4.5. Çok Katmanlı Perseptron ve Geri Yayılımlı Öğrenme 46
4.6. Takviyeli Öğrenmede Yapay Sinir Ağları 47
5. Yapay Sinir Ağıyla Hızlandırılmış Takviyeli Öğrenme 49
5.1. Etmen ve Çevrenin Özellikleri 49
5.1.1. Çevrenin özellikleri 49
5.1.2. Etmenin özellikleri 51
5.2. Sinir Ağıyla Hızlandırılmış Takviyeli Öğrenme Mimarisi 53
5.2.1. Eylem seçme 57
5.3. Sinir Ağıyla Hızlandırılmış Takviyeli Öğrenme Algoritması 57
5.4. Bilgisayar Simülasyonu ve Deneysel Sonuçlar 59
5.4.1. Bilgisayar simülasyonu 59
5.4.2. Deneysel Sonuçlar 59
5.4.2.1. Deneysel Örnek – 1 60
5.4.2.2. Deneysel Örnek – 2 61
5.4.2.3. Deneysel Örnek – 3 62
6. SONUÇ 63
KAYNAKLAR 64

1. GİRİŞ

1.1. Problemin Tanımı

Dağıtık yapay zeka 1970'li yılların ortasından itibaren hızla gelişmiş ve çeşitlenmiştir. Günümüzde doğruluğu kabul edilen ve ilerisi için ümit verici bütün araştırma ve uygulamalarda kullanılan kavram, fikir ve algoritmalar yapay zekanın birer ürünüdür. Yapay zeka; bilgisayar bilimleri, ekonomi, organizasyon, yönetim bilimi ve psikoloji gibi birçok ana bilim dalında kullanılmaktadır[1].

Dağıtık yapay zeka çalışmalarının birçoğu, akıllı etmenin hedef veya amaçlarını gerçekleştirmek için çevresiyle etkileşim içinde bulunarak oluşturdukları bağımsız etmen uygulamalarının birer ürünüdür.

Takviyeli öğrenme, akıllı etmen sistemlerinde öğrenme için güçlü bir yöntem olarak önerilmiştir. Takviyeli öğrenme, bağımsız bir etmenin etrafını algılayarak hedefine ulaşması için optimal hareketleri seçmesini sağlar.

Takviyeli öğrenme metotları, hakkında bilgi sahibi olamadıkları dinamik veya belirsiz ortamlarda uygulanabilirler. Takviyeli öğrenme metotları çevresiyle etkileşim içinde bulunarak tecrübesini arttırır. Sürekli ödül alan hareketleri seçerek başarısını maksimum değere çıkarmaya çalışır. Bu nedenle dünya üzerindeki genel problemlerin çözülmesinde kullanılabilir.

Takviyeli öğrenme metotları kullanılarak bir robota etrafındaki engellere takılmadan hareket etmesi öğretilebileceği gibi akıllı bir etmenin internet üzerinde filtreleme ve arama yaparak kullanıcı tarafından belirlenen dokümanları bulması da sağlanabilir.

1.2. Amaç

Bu çalışmada yapay sinir ağları kullanılarak hızlandırılmış takviyeli öğrenme metodu geliştirilmiştir. Takviyeli öğrenme, eğitim örneklerine sahip olunamayan dinamik çevrelerde etmenin deneme yanılma yoluyla çevre içerisinde nasıl davranacağını öğrenmesidir[2]. Q öğrenme, takviyeli öğrenme metotları içinde en popüler algoritmalardan bir tanesidir. Yapay sinir ağları da takviyeli öğrenme gibi ortam hakkında fazla bilgi sahibi olunamayan gerçek çevrelerde uygulanabilen makine öğrenmesi konularındandır.

Çalışmamda bu iki metot birleştirilerek, dinamik bir çevreye yerleştirilen etmenin engellerden kaçınarak çıkışı bulması amaçlanmıştır. Etmenin çevre içerisindeki hareketlerinde q değerlerinin yanı sıra yapay sinir ağı çıkışlarından da faydalanılmıştır. Böylece bağımsız etmenin karmaşık çevrelere daha çabuk adapte olması ve daha kısa sürede hedefine ulaşması sağlanmıştır. Deneysel sonuçlar gösteriyor ki yapay sinir ağları kullanılarak hızlandırılmış takviyeli öğrenme metodu etmenin optimum yolu daha çabuk bularak hedefine ulaşmasını kolaylaştırmaktadır.

2. LİTERATÜR ARAŞTIRMASI

Çoklu etmen sistemler, takviyeli öğrenme ve q öğrenme üzerinde bir çalışma yapılmıştır. Ortam içerisine çeşitli algılayıcılara sahip av ve avcılar konulup, takviyeli öğrenme metotları kullanılarak bir arama problemi oluşturulmuştur. Çoklu etmen sistemler için önemli bir konu olan davranış ve çaba üzerinde durulmuştur. Etmenler arasında ortak bir davranış biçimi geliştirilmesi, bir etmenin tek başına başaramayacağı görevi birden fazla etmenin ortak işbirlikçi hareketlerle gerçekleştirebileceği üzerinde bir çalışma yapılmıştır. (Ishiwaka, Y., Takamasa, S., Kakuza, Y., 2003)

Çoklu etmen sistemler ve takviyeli öğrenme teknikleri üzerine bir çalışma yapılmıştır. Çoklu etmen sistemler için takviyeli öğrenme metotlarının ne kadar önemli olduğu vurgulanmıştır. Günümüze kadar gerçekleştirilen çeşitli takviyeli öğrenme metotları arasında karşılaştırma yapılmıştır. (Peeters, M., Verbeeck, K., Nowe, A., 2003)

Takviyeli öğrenme metotlarının çok geniş alanlarda uygulanması üzerine bir çalışma yapılmıştır. Takviyeli öğrenme algoritmasının çok geniş durum ve eylem kümesine sahip olduğu durumlarda öğrenmeyi ve hareket seçimini hızlandırmak için bulanık mantıktan faydalanılmıştır. Öğrenme algoritması önüne bir değerlendirme fonksiyonu konularak, seçilebilecek durum ve eylem kümeleri basitleştirilip etmenin çevre içerisinde daha kolay bir şekilde hareket etmesi sağlanmıştır. (Hoshino, Y., Kamei, K., 2003)

Robotların çevre içinde engellere takılmadan hareket etmeleri üzerine bir çalışma yapılmıştır. Takviyeli öğrenme ve yapay sinir ağları algoritmaları kullanılmıştır. Çevre içinde bulunan etmenin, engelleri algılayıcıları vasıtasıyla algılayarak, engellere takılmadan hareket etmesi sağlanmıştır. (Huang, B., Cao, G., Guo, M., 2005)

Bağımsız etmenler ve modüler takviyeli öğrenme teknikleri ile mobil robot tasarlama üzerine bir çalışma yapılmıştır. Etmenin içinde yaşamış olduğu ortamı q değerlerine göre birden fazla alana ayırarak, daha küçük alt ortamlarda öğrenme yapılmıştır. Bu yöntemle, öğrenme daha hızlı ve verimli bir hale getirilmiştir. (Uchibe, E., Minoru, A., Hosoda, K., 1996)

Robot tasarımı ve takviyeli öğrenme metodu üzerinde bir çalışma yapılmıştır. Genellikle takviyeli öğrenme çalışmalarının yüksek maliyet ve gerçek ortam koşullarının zorluğu yüzünden simülasyon ortamlarda tasarlandığından bahsedilmiştir. Robot tasarımı yapılarak, takviyeli öğrenme teknikleri yardımıyla robotun gerçek ortam şartlarına uyum sağlaması üzerinde durulmuştur. (Yamaguchi, T., Masubuchi, M., Fujihara, K., 1996)

Paket yönlendirme ve q öğrenme algoritması üzerine bir çalışma yapılmıştır. Bir ağ içerisine ortamı dinleyen ve paket sürelerini hesaplayan bir takviyeli öğrenme modülü yerleştirilmiştir. Ağdaki yoğunluk hesaplanarak paketlerin gidiş yörüngeleri takviyeli öğrenme modülü yardımıyla belirlenmiştir. (Boyan, J., Litmann, M., 1998)

Takviyeli öğrenme ve grid hesaplamaları üzerine bir çalışma yapılmıştır. Ortamdaki yapılması gereken işler, bu işleri gerçekleştirecek olan kaynakların ortam içine yerleştirilmesi ve daha fazla verim alınması için gerekli hesaplamalar takviyeli öğrenme metotları kullanılarak gerçekleştirilmiştir. (Galstyan, A., Czajkowski, K., 2004)

3. ETMEN VE TAKVİYELİ ÖĞRENME

3.1. Giriş

Eğer uygun yazılımlar geliştirilebilirse dünya üzerinde birçok çözülemeyen problem bilgisayarlar tarafından çözülebilir. Uçaklar için uçuş kontrol sistemleri, otomatik imalat sistemleri ve lineer olmayan sistemler buna örnek olarak verilebilir. Bu gibi problemlerin bir çoğu günümüzdeki bilgisayarların çok yavaş kalması ve hafızalarının yeterli olmaması sebebiyle çözülememektedir. Ama asıl zorluk bu gibi durumları çözebilecek programların olmamasıdır. Eğer bir bilgisayar deneme yanılma yoluyla öğrenebiliyor ve problemi çözebiliyorsa bu program geliştirme için önemli bir gelişmedir[3].

Bütün takviyeli öğrenme metotları için ortak hedef, etmenin deneme yanılma tekniğini kullanarak çevresiyle etkileşim içerisine girip kendisini amacına ulaştıracak sıralı eylem kümesini belirlemektir[4].

3.2. Öğrenme Metotları Arasındaki Farklar

Günümüze kadar geliştirilen öğrenme metotlarını üç ana grupta sınıflandırabiliriz:

- **Planlı öğrenme (Planning Learning):**
  - Etmen başlangıç aşamasından amacına ulaşıncaya kadar önceden planlanmış bir algoritmaya göre hareket eder.

- Etmenin gerçekleştireceği durum-eylem çiftleri önceden belirlenmiştir. Etmen bulunduğu durumda hangi hareketi gerçekleştireceğini ve hangi duruma geleceğini daha önceden bilir.
- Termostatla çalışan bir klimanın oda sıcaklığını sabit bir derecede tutmayı öğrenmesi buna örnek olarak verilebilir.

- **Denetleyerek Öğrenme (Supervised Learning):**

  - Etmene yapması gerekenleri gösteren eğitim örnekleri verilir. Etmen bu örnekleri kullanarak kendini eğitir.
  - Eğitim aşaması tamamlandıktan sonra etmenin ortam içindeki yaşam süreci başlatılır.
  - Geri beslemeli yapay sinir ağları algoritması, bir denetleyerek öğrenme metodudur.
  - Eğitilmiş bir robotun, çöp kutularını eğitim örneklerinden tanıyarak, bir oda içindeki çöpleri toplaması örnek olarak verilebilir.

- **Takviyeli Öğrenme (Reinforcement Learning):**

  - Robot bulunduğu durum ve gerçekleştireceği eylem hakkında herhangi bir bilgiye sahip değildir.
  - Robot gerçekleştirdiği eylemler sonrası çevreden aldığı ödüllerle kendini eğitir. Ödül, robotun amacına ulaşması için gerçekleştirdiği eylemin ne kadar önemli olup olmadığını gösteren sayısal bir değerdir.
  - Bir labirent ortamına bırakılan robotun deneme yanılma yoluyla, çıkışı en kısa mesafeden bulmayı öğrenmesi bu algoritmaya örnek olarak verilebilir.

3.3. Etmen Ne Demektir?

Etmen belirli bir çevreye yerleştirilmiş bir bilgisayar sistemidir. Etmen bir yazılım ürünü olan program olabileceği gibi bir donanım ürünü olan robot da olabilir[5]. Etmen, çevresinde gelişen olayları algılayıcıları vasıtasıyla algılayarak bağımsız çıkış hareketleri üretir. Böylece çevresindeki elemanlarla etkileşim içinde bulunarak yaşam sürecini devam ettirir. Buna örnek olarak bir robotun fabrikadaki işlemleri optimize etmeyi öğrenmesi ve oyun oynamayı öğrenmesi verilebilir.

Etmen esnek ve mantıklı davranabilme özelliğinden dolayı çevresinde gelişen durumları anlayabilen bir varlıktır. Mantıklı ve esnek davranabilmesi, etmenin problem çözme, planlama, karar verme ve öğrenme gibi önemli görevleri başarıyla yapmasını sağlar. Etkileşim içindeki etmen, çevresindeki diğer etmenlerden veya insanlardan etkilenebilir. Etmenler ortak bir görevi başarabilmek için işbirliği içinde olabilecekleri gibi bazı bireysel etmenler diğer etmenlerle rekabete de girebilir.

Dağıtık yapay zekada temel amaç, etmenlerin insanlar gibi birbirleriyle etkileşim içinde bulunarak birbirlerini en iyi şekilde anlamalarını sağlamaktır. Çünkü etmen mimarisi günümüzde ve gelecekteki bilgisayar bilimleri ve uygulamalarında önemli görevler almaktadır.

Modern sistemleri sıkıntıya sokan geniş çevre uzayı, büyük veriler ve işlem yoğunluğundan dolayı oluşan karmaşık yapıların çözümünde etmen mimarisi önemli rol oynamaktadır[1]. Etmenleri diğer öğrenme algoritmalarından ayıran en önemli özellik, birçok durum uzayına kolaylıkla uyum sağlayabilmeleri ve öğrenmek için herhangi bir ön bilgiye ihtiyaç duymamalarıdır.

3.4. Etmenin Özellikleri

3.4.1. Zeka

Etmenler insanlara mahsus özellikler taşırlar. Mesela; basit bir etmen, bir malın fiyatı belli bir ücretin altına düştüğü zaman bu malı satın alabilir. Bir internet etmeni bir aramayı birden fazla site üzerinden yaparak sonuçları birleştirebilir. Etmenler kendi deneyimlerinden, diğer etmenlerden, kullanıcılardan ve bulundukları çevreden faydalanarak öğrenirler[6].

3.4.2. Özerklik

Zekaya ek olarak etmenler için önemli bir özellik de özerkliktir. Özerklik, etmenin programcıdan veya kullanıcıdan bağımsız olarak kendi kendine karar verebilme özelliğidir. Bu özelliğe örnek olarak, alışveriş yapan bir etmenin kullanıcıya sormadan kullanıcının ihtiyacı olan bir malı bilmesi ve satın alması gösterilebilir[6].

3.4.3. Öğrenebilme yeteneği

Etmen öğrenme yeteneğine sahiptir. Başka bir deyişle; etmen yaptığı hareket sonrası çevre tarafından verilen yeni bilgileri uygun bir formda saklar. Öğrenen etmenler yaşamları boyunca gerçekleştirdikleri görevlerle performanslarını arttırırlar. Eğer bir etmen başarısızlıkla sonuçlanan bir hareket gerçekleştirirse bu, etmenin tecrübelerinden yararlanarak aynı hatayı ileride tekrar yapmasını engeller[6].

3.4.4. İş birliği, ortak çalışma

Çoklu etmen sistemlerinde, etmenler genellikle diğer etmenlerle ortaklaşa çalışırlar. Mesela; bir arama problemindeki etmen, diğer arayıcı etmenlerle işbirliği içinde çalışarak, bulmaya çalışılan nesnenin yerini daha çabuk tespit edebilir[6].

3.4.5. Etmenin diğer özellikleri

Hareketlilik : Etmenin elektronik ağlar üzerinde hareket etme yeteneğidir. Yani etmen, kullanıcının belirlediği işi yapmak için bir yerden başka bir yere gidebilir. Bunun dışında mekanik aksama sahip etmen, hedefine ulaşabilmek için ortam içinde ileri gitmek, sağa gitmek, sola gitmek, geri gitmek gibi hareketlerde bulunabilir.

Doğruluk : Bazı etmenler yetenek ve tecrübelerini geliştirebilmek için diğer etmenlere yalan söyleyebilir, oysa diğer etmenler sürekli doğruyu söyler.

Yardımseverlik : Etmenler birbirlerine ters düşen hedef veya amaçlara sahip değillerdir. Etmenler amaçları doğrultusunda sürekli birbirleriyle yardımlaşırlar.

Mantıklılık : Etmenler yaşamları boyunca sahip oldukları çevreleri algılar, zaman içerisinde çevreleriyle etkileşim içine girerler. Bunun sonucunda etmenler amaçlarının gerçekleştirilmesini önleyen engellerden kaçınır ve kendilerini aktive ederler.

3.5. Etmen Sınıflandırması

Etmenleri temel olarak altı alt grupta sınıflayabiliriz:

- Tepkisel etmenler
- İşbirlikçi etmenler
- Arabirim etmenler
- Mobil etmenler
- Bilgilendirme Etmenleri
- Melez Etmenler

3.5.1. Tepkisel etmenler

Tepkisel etmenler, çevreden aldıkları giriş değerlerini daha önceden belirlenen kurallarla karşılaştırıp, çıkış olarak bir eylem gerçekleştirirler. Tepkisel etmenlere örnek olarak mail sunumcularında kullanılan e-mail filtreleme sistemleri verilebilir. E-mail filtreleme sistemleri aldığı her maili daha önceden belirlenen kurallara göre sınıflandırabilir. E-mail sunumcularında ortak olarak kullanılan sınıflandırma, bu tip e-mailleri “spam olarak kabul et”, “reddet” veya “junk mail olarak işaretle”dir. E-mail filtreleme sistemi için çevre, gelen e-mailler ve kullanıcıların e-mail hesaplarıdır[6].

3.5.1.1. Hedefe yönelik etmenler

Tepkisel etmenlerden daha karmaşık bir yapıya sahiptirler. Hedefe yönelik etmenler, önceden tanımlanan kurallardan bağımsız bir hedefi gerçekleştirmeye çalışırlar. Bu işlemi arama yaparak veya belirli planlar dahilinde gerçekleştirirler. Mesela; bir etmenin görevi internette yapay zeka ile ilgili yapılan çalışmaları bulmak olabilir. Bu durumda etmenin görevi web sayfalarını bulmak, onları yüklemek ve yeni açılan sayfalardaki linkleri takip etmek olabilir. Etmen sonuç olarak kendi kriterlerine göre bulduğu sayfaları kullanıcısına göstererek hedefine ulaşmış olur[6].

3.5.1.2. Faydaya yönelik etmenler

Faydaya yönelik etmenler hedefe yönelik etmenlere benzerler fakat amaçlarına ulaşmalarına ek olarak sürekli elde ettikleri toplam fayda değerlerini maksimuma çıkartmaya çalışırlar. Fayda değeri, etmenlerin ne kadar başarılı olduğunu gösterir. Etmenin hedefine ulaşması için ne kadar daha çalışması gerektiği bu değer vasıtasıyla hesaplanabilir[6].

3.5.2. İşbirlikçi etmenler

İşbirliği yapan etmenler kullanıcıları adına işlerini yürütürken özerk çalışma ve diğer etmenlerle çalışma özelliği taşırlar. Birden fazla amacı olan etmenin, işlerini gerçekleştirebilmesi için, işle ilgili çok geniş bir bilgi birikimine ve karmaşık bir fonksiyon yapısına sahip olması gerekir. Bir etmene aynı anda birden fazla görev yüklemek etmenin verimliliğini azaltır. Bunun yerine amaca uygun bilgiyi bulmak ve diğer görevleri yerine getirmek için işbirliği yapan çoklu etmen sistemleri kullanmak daha doğru bir çözümdür[6].

3.5.3. Arabirim etmenler

Arabirim etmenler kişisel asistanlar olarak düşünülebilir. Arabirim etmenler tıpkı bağımsız etmenler gibi yaptıkları görevlerden öğrenme yeteneklerini geliştirirler. Arabirim etmenler insanlarla işbirliği içindedirler fakat diğer etmenlerle işbirliği yapmalarına gerek yoktur. Bazı durumlarda arabirim etmenler diğer etmenlerin deneyimlerinden faydalanarak öğrenebilirler[7].

Arabirim etmenler görevlerini gerçekleştirmek için gerekli olan bilgiyi kullanıcıdan öğrenerek hareket ederler. Etmen, arabiriminde kullanıcı tarafından yapılan

hareketleri izler. Bir anlamda kullanıcısının omuzlarında olup biteni takip eder ve sahibi için bu işi yapmanın daha kolay bir yolunu bulmaya çalışır.

Kozierok and Maes (1993) takvim üzerinde, programlı buluşmaları ayarlayan bir arabirim etmen tasarlamıştır. Bu etmen, kullanıcısı adına diğer kişilerle buluşmaları düzenler, kişileri reddeder veya planlanmış bir buluşma üzerinde değişiklik yapabilir. Etmen kullanıcısını gözlemleyerek hangi günler ne tür buluşmalar yaptığını öğrenir ve o gün için buluşmalarını ona göre düzenler. Temel olarak arabirim etmenler, kullanıcısı için öğrenmeyi dört yolla gerçekleştirir:

- Kullanıcıyı izleyerek ve takip ederek (Kullanıcıdan öğrenme)
- Kullanıcıdan pozitif veya negatif bir geri besleme alarak (Kullanıcıdan öğrenme)
- Kullanıcıdan açık talimatlar alarak (Kullanıcıdan öğrenme)
- Tavsiye için diğer etmenlere sorarak (Diğer etmenlerden öğrenme)

3.5.4. Mobil etmenler

Mobil etmenler bir yerden başka bir yere gitme yani hareket etme özelliğine sahip etmenlerdir. Mobil etmen yazılımları internet veya bilgisayar ağları üzerinde sürekli hareket ederler. Mobil etmenler bir bilgisayardan başka bir bilgisayara hareket ederek sahibi adına gerekli olan bilgileri toplar, yapması gereken eylemleri gerçekleştirir ve kullanıcısına geri döner[7].

Bilgisayar virüsleri mobil etmen olarak tanımlanabilirler. Çünkü virüsler kullanıcıdan talimatlar almadan, bağımsız olarak hareket edebilirler. Kendilerini bilgisayar ağları üzerinden diğer bilgisayarlara taşıyabilir, elde ettikleri bilgileri merkez bilgisayarlara taşıyabilir veya bilgileri silebilirler.

Etmen tasarımcıları, mobil etmenlerin bir ağda gezinmesi ve etmen davranışlarını tanımlamaları için bir program diline ihtiyaç duyarlar. Bu etmen haberleşme dillerinden bazıları şunlardır:

Tcl ve Safe Tcl: Bir e-mail mesajının içeriği gibi, etmen işlemlerini nakletmek için e-mail kullanan makineden bağımsız bir script dilidir. Küçük ölçekli ticari uygulamaları geliştirmek için kullanılır.

Telescript: General Magic firması tarafından hareketli etmenler için tasarlanan bir etmen çalışma ortamıdır. Kablosuz ağ ortamlarındaki ticari uygulamalar için java programlama dilinde yazılmıştır.

Mobil etmenlerin temel avantajı verimli olmalarıdır. Bir etmen aynı anda birden fazla uzak bilgisayara bağlanabilir, yüksek miktardaki verileri karar vermek için bilgisayar ağları üzerinden alabilir. Bunun yanında uzak bilgisayardaki kullanıcılar için gerekli olan bilgileri merkez bilgisayardan sorgulayabilir. Bu işlemleri gerçekleştiren mobil etmenlerin ana faydaları şöyledir:

- Haberleşme masraflarını azaltır
- Koordinasyon kolaylığı sağlar
- Esnek ve dağıtık bir yapı oluşturur

3.5.5. Bilgilendirme etmenleri

Bilgilendirme etmenleri genellikle internet üzerinde kullanılırlar ve internet etmenleri olarak çağrılabilirler. Bir bilgilendirme etmeni internet üzerindeki çok geniş kaynaklar arasında kullanıcının istediği verileri bulmasını, filtrelemesini ve sınıflandırmasını sağlar.

Bilgilendirme etmenleri, bilgi sağlayıcıları tarafından yayınlanmış çevrim içi bilgiyi depolayan sunumcular üzerinde bulunurlar. Microsoft Internet Explorer, Mozilla, Netscape Navigator gibi istemci tarayıcı etkileşimli bir şekilde kullanıcılar tarafından erişilebilirler. Bir internet sunumcusu, http portları üzerinden yayın yapan ve internet üzerinden gelen istemcilerin isteklerini karşılayan bir sunumcu programıyla birlikte Solaris, Unix, Linux, Macos veya Microsoft Windows Server 2003 ortamında çalışır. Bilgilendirme etmenleri bu sunumcularda bulunan bilgisayar programlarıdır ve onlar doğrudan kullanıcı müdahalesi olmadan internet üzerindeki dağıtık bilgiye erişebilirler[7].

İnternet etmenleri temel olarak aşağıdaki etmen çeşitlerinden meydana gelir:

Web Arama Etmenleri : Günümüzde Google, Yahoo, Alta Vista gibi arama motorları arama servisliği sağlayan başlıca arama motorlarıdır. Tipik olarak; kullanıcı, arama motoruna arayacağı konu ile ilgili anahtar kelimeleri girer ve arama motoru da bu kelimeleri sorgulayarak buna uyan URL'lerin (Uniform Resource Locations) bir listesini geri gönderir.

Bilgi Filtreleme Etmenleri : Bilgi filtreleme etmenleri kullanıcı tarafından verilen anahtar kelimeleri, web sayfaları veya haber kaynakları gibi değişik kaynaklardan toplayarak, uygun bir filtreleme gerçekleştirdikten sonra bir Web sayfası veya e-posta ile kullanıcısına sunar.

Bildirim Etmenleri : Bildirim etmenleri önemli olan olayları kullanıcısına veren etmen çeşididir. Burada olaylar bilginin durumunda meydana gelen değişikliklerdir. Bunlar aşağıdaki gibi olabilir:

- Belirli bir web sayfasında meydana gelen değişiklik

- Belirli anahtar kelimelerle yapılan sorgulamada arama makinesine ilave edilen dokümanlar
- Doğum günü gibi kullanıcıya ait şahsi olaylar

Diğer internet etmenleri çeşitleri; haber verici etmenler, kitap bulucu etmenler, iş bilgi izleme etmenleri, sınıflandırıcı etmenler, mektup etmenleri, mail servis etmenleri, iş etmenleri, eğlence etmenleri ve alış veriş etmenleri olarak belirlenebilir.

3.5.6. Melez etmenler

Şimdiye kadar etmenlerin beş tipini inceledik. Bunlar işbirlikçi, tepkisel, arabirim, mobil ve bilgilendirme etmenleridir. Bunların hangisinin daha iyi olduğunu söylemek zordur. Çünkü her bir tipin kuvvetli olduğu ve eksik olduğu yanları vardır. Bu sebeple kullanıcı için amaç, ilgili tekniğin kuvvetli olan yanlarını arttırmak ve zayıf olan yanlarını azaltmaktır. Çoğu kere bunu yapmanın tek yolu ise melez yaklaşımı kullanmaktır. Melez yaklaşım etmenlere sağlamlık, daha hızlı cevap verme, çevreye ve şartlara uyma yeteneklerini de kazandırır[7].

3.6. Etmen ve Çevre

Çevre içinde bulunan etmenin amacı, çevre ile etkileşim içine girerek toplam ödül miktarını maksimuma çıkaran hareketlerin sırasını öğrenmektir. Bu amaçla üretilen bir robotun, çevrenin durumunu gözleyebilmek için algılayıcılara ve bu durumu değiştirmek için de bir hareket kümesine sahip olması gerekmektedir. Örnek olarak etmen hareketli bir kameraya, algılayıcılara ve ileri gitmek, geri dönmek, sağa gitmek, sola gitmek gibi hareketlere sahip olabilir. Etmenin görevi, amacını gerçekleştirebilmek için bir kontrol stratejisi veya politikası öğrenmektir.

Bu kontrol stratejisi ve karar verme işlemindeki karmaşıklık değişik çevre koşulları tarafından da etkilenir. Russell ve Norvig çevrenin özelliklerini beş ana grupta sınıflandırmayı önermişlerdir[8]:

- **Ulaşılabilen ve Ulaşılamayan Çevre:** Ulaşılabilen çevre etmenin tamamen eline geçirebileceği çevredir. Bu sayede etmen çevreden tüm güncel bilgileri elde edebilir. Fiziksel dünya ve internet gibi karmaşık çevreler de ulaşılamayan çevre olarak tanımlanabilir. Etmen, ulaşılabilen çevrelere ulaşılamayan çevrelerden daha çabuk adapte olur ve öğrenme aşamasını daha hızlı gerçekleştirebilir.

- **Belirgin ve Belirgin Olmayan Çevre:** Belirgin olan çevrede etmen gerçekleştireceği olayın sonucu hakkında hiçbir belirsiz bilgiye sahip değildir. Fiziksel dünya belirsiz bir çevre olarak tanımlanabilir. Belirsiz çevreler etmen tasarımında büyük problem oluşturabilir.

- **Statik ve Dinamik Çevre:** Statik çevrede etmen tarafından gerçekleştirilen olaylar değişmeden kalır. Dinamik çevrede ise araya başka olaylar girer. Bu yüzden çevre, etmenin kontrolünden çıkabilir.

- **Kesikli ve Sürekli Çevre:** Eğer etmen sabit sayıda eylem gerçekleştiriyorsa çevre kesiklidir. Eğer etmen çevreyle sürekli etkileşim içinde ve sonsuz sayıda eylem gerçekleştiriyorsa çevre süreklidir.

- **Episodik ve Episodik Olmayan Çevre:** Episodik çevre etmenin farklı yollardan amacına ulaşabildiği ve en uygun yolu bulabildiği çevredir. Episodik olmayan çevrede ise etmen amacına sadece tek bir yol üzerinden ulaşabilir ve bu yol da etmen için optimum yoldur.

Etmenlerin başarısı, içinde bulundukları ortamların karmaşıklığıyla ters orantılıdır. Ortam ne kadar karmaşıksa etmenin öğrenmesi o kadar zordur. Russell ve

Norvig'e göre en karmaşık çevreler ulaşılamayan, belirgin olmayan, episodik olmayan ve sürekli olan çevrelerdir.

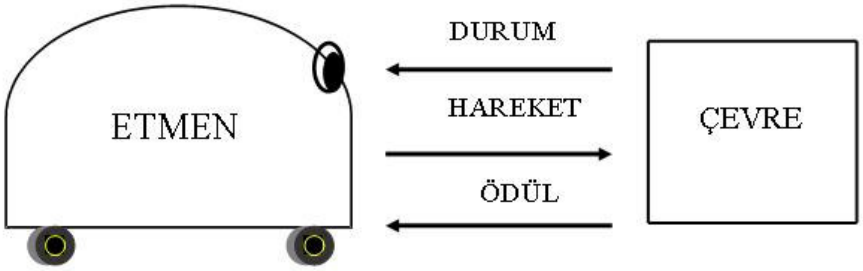
Etmenlerin içinde bulunabileceği çevre örnekleri ve özellikleri Tablo 3.1'de gösterilmektedir.

Ortam	**Ulaşılabilir**	**Belirgin**	**Statik**	**Kesikli**	**Episodik**
Saate karşı satranç oynayan etmen	✓	✓	Yarı	✓	X
Saatsiz satranç oynayan etmen	✓	✓	✓	✓	X
Poker oynayan etmen	X	X	✓	✓	X
Tavla oynayan etmen	✓	X	✓	✓	X
Taksi kullanan etmen	X	X	X	X	X

Tablo 3.1 Etmenin içinde bulunduğu çevre örnekleri ve özellikleri [9]

3.7. Çevresiyle Etkileşen Etmen

Eğitim örnekleri bulunmayan ortamlarda etmen çevresiyle etkileşim içine girerek öğrenir. Etmen algılayıcıları vasıtasıyla çevreden durum bilgisini öğrenir. Bu bilgi sonrasında çevreyi etkileyen bir hareket üretir. Bu hareket sonrasında eğitici veya yazılım tarafından belirlenen bir ödül ile ödüllendirilir.



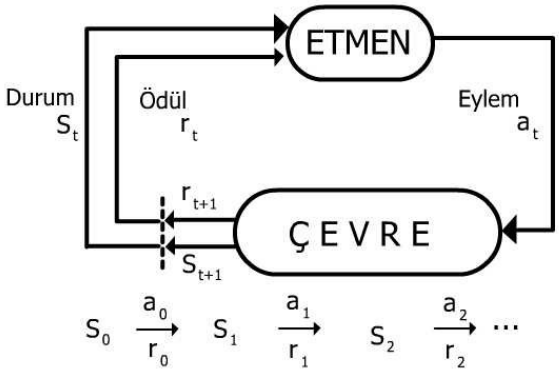
Şekil 3.1 Çevresiyle Etkileşen Etmen

3.8. Takviyeli Öğrenme

Takviyeli öğrenme, kendi ortamını algılayan ve ortamında hareket yapan özerk bir etmenin, amacını gerçekleştirmek için en uygun hareketleri yapmayı nasıl öğrenebileceği sorusuna cevap verir[10]. Bu yöntem robotik, oyun programlama, hastalık teşhisi ve fabrika otomasyonu gibi birçok dalda yaygın olarak kullanılmaktadır.

Takviyeli öğrenme, bir çeşit denetlenmeyen öğrenme (unsupervised learning) sistemidir. Etmen, ortamında bir hareket yaptığı zaman, eğitici veya yazılım, yeni durumun istenilebilirliğini göstermek için bir ödül veya ceza ile etmeni takviye eder. Etmen, amacına ulaşmak için gerçekleştirebileceği en iyi eylemi seçmeye çalışır[2]. Mesela; oyun oynayan bir etmen eğitileceği zaman, eğitici veya yazılım, oyun kazanıldığında pozitif bir ödül, kaybedildiğinde negatif bir ödül ve diğer durumlarda sıfır ödül sağlayabilir. Etmenin amacı, en büyük toplam ödülü üreten hareketlerin sırasını öğrenmektir.

Standart takviyeli öğrenme sisteminde etmen, Şekil 3.2'de gösterildiği gibi ortamla algı ve hareket mekanizmaları sayesinde etkileşir.



Şekil 3.2 Takviyeli Öğrenme Sistemi

Böyle bir model aşağıdaki bileşenleri kapsar:

- Etmen ve çevre birbiriyle farklı birçok zaman adımında etkileşir.

  $t = 0, 1, 2, 3, \ldots$

- Her zaman adımında çevrenin durumu $s_t \in S$ şeklinde gösterilir. S çevrenin bulunabileceği bütün durumları kapsayan bir kümedir.

  $S = \{s_1, s_2, s_3, \ldots\}$

- Her durumda etmen çevresiyle etkileşerek bir eylem gerçekleştirir.

  $a_t \in A(s_t)$

- $A(s_t)$ kümesi etmenin s_t durumunda gerçekleştirebileceği eylemlerin kümesidir.

  $A(s_t) = \{a_1, a_2, a_3, \ldots\}$

- Bir adım sonra, etmen gerçekleştirdiği eylemin sonucunu çevreden alır. Etmen, gerçekleştirdiği durumun iyiliğini gösteren sayısal bir değeri ödül olarak alır ve kendisini yeni bir konumda bulur.

  R: SxA → R

Şekil 3.2'de görüldüğü gibi, toplam ödül miktarını maksimuma çıkarmak için bir kontrol politikasını öğrenme problemi çok geneldir ve robotun öğrenmesinin dışında birçok problemi kapsar. Problem genelde gerçekleştirilecek eylemlerin sırasını bulmaktır. Mesela; üretim optimizasyon problemlerinde öyle üretim hareketleri seçilmelidir ki maksimize edilecek ödül, üretilen malın satış değerinden mal olma maliyetinin çıkarılması sonucu elde edilen değerdir. Başka bir örnek olarak da planlama problemleri gösterilebilir. Büyük bir şehirdeki yolculara hangi taksilerin gönderileceğini seçmede maksimize edilecek ödül, yolcuların bekleme zamanına ve hareket halinde iken taksilerin harcayacakları toplam benzin miktarına bağlıdır. Amaç, gerçekleştirilen hareketin sırasını ve kalitesini hesaplamaktır. İstenilen duruma erişmek için arama yöntemini kullanan bir sistem, her bir adımda alternatif hareketler arasında bir seçim yapar. Takviyeli öğrenmedeki amaç, etmenin istenilen duruma en kısa sürede gitmesini sağlamaktır. Takviyeli öğrenme ile yapılan uygulamaların belki de en tanınmışı Tesauro' nun (1995) TD-GAMMON oyun programıdır. Bu program birinci sınıf tavla oyuncusu olmak için takviyeli öğrenme metodunu kullanmıştır[11].

Takviyeli öğrenme ile problemleri çözmek için iki ana yöntem vardır: Bunlardan ilki, ortamı iyiye götüreni bulmak için eğitim uzayında bir arama yapmak, ikincisi ise faydalı hareketi tahmin etmek için istatistik ve dinamik programlama yöntemlerini kullanmaktır[12].

Takviyeli öğrenme problemi için önerilen diğer önemli yaklaşımlar modüler ve fonksiyon yaklaştırma yöntemleridir. Uygun hareket seçmek için bir kontrol politikası

öğrenme problemi, fonksiyon yaklaştırma problemlerine benzerlik gösterir. Öğrenilmesi gereken hedef fonksiyon, bir kontrol politikasıdır.

$$\Pi : S \rightarrow A$$

Denklem 3.1 Politika

Kontrol politikası etmene S kümesinden şu anki s durumu verildikten sonra A kümesinden uygun bir hareket a'yı çıkarma politikasıdır. Bununla birlikte takviyeli öğrenme birkaç yönden diğer fonksiyon yaklaşımlı yöntemlerden ayrılır(Mitchell, 1977).

- **Gecikmiş Ödül:** Etmenin amacı, şu andaki s durumundan, en uygun hareket a= π (s)'i planlayan bir hedef fonksiyon öğrenmektir. Diğer birçok öğrenme yöntemlerinde π gibi bir hedef fonksiyon öğrenildiği zaman her bir eğitme örneği (s, π (s)) şeklindedir. Takviyeli öğrenmede ise eğitme bilgisi bu formda değildir. Bunun yerine eğitici, etmen hareketlerini yürütürken sadece anlık ödül değerlerinin sırasını sağlar[11].

- **Kısmen Gözlenebilir Durumlar:** Her ne kadar herhangi bir zaman adımında etmen algılayıcılarının ortamın bütün durumunu algılayabildiğini kabul etmek uygun olsa da, birçok pratiksel durumda algılayıcılar yeterince bilgi sağlayamaz. Örneğin; sadece önünü görmeye yarayan kameraya sahip bir robot, arkasında ne olduğunu idrak edemez. Böyle durumlarda hareketler seçildiği zaman, şu andaki gözlemlerle birlikte öncekileri de dikkate almak etmen için gerekli olabilir[11].

- **Hayat Boyu Öğrenme:** Birbirinden izole edilmiş fonksiyon yaklaşımından farklı olarak, robot öğrenme aşamasındayken aynı algılayıcıları kullanarak aynı ortamda birden fazla ilgili işi öğrenmeyi ihtiva edebilir. Örneğin; hareketli bir robotun aynı anda şarj edicisine yaklaşma, dar koridorlar

boyunca gezinme ve lazer yazıcıdan çıktıları toplama gibi öğrenme amaçları olabilir[11].

3.9. Takviyeli Öğrenme Algoritması

Takviyeli öğrenme ortamın dinamikleri hakkında hemen hemen hiçbir şey istemeyen bir tekniktir. Küçük çocuklardaki gibi hatalardan ders çıkartılması üzerine kurulmuştur. Bir ortamda görevlendirilmiş bir etmen amacını bilir ve içinde bulunduğu durumu amacını da içeren bir duruma nasıl çevirmesi gerektiğini öğrenir[13].

3.9.1. Amaç ve ödül

Takviyeli öğrenmede etmenin amacına ulaşması için eğitmen veya yazılım tarafından belirlenen amaç ve ödüller büyük önem taşımaktadır.

Mesela; bir labirentten çıkmaya çalışan bir robotu düşünelim. Birinci aşamada robotun çıkışı bulma hareketine +1, diğer durumlara da 0 ödülü verilsin. İkinci aşamada ise robotun çıkışı bulma hareketine +1, diğer durumlara ise -1 ödülü verilsin. Bu iki öğrenme metodu incelendiğinde ve robotun asıl amacının en yüksek ödülü toplamak olduğu düşünüldüğünde, robotun ikinci yöntemde çıkışı daha çabuk bulması olasıdır.

Bu nedenle takviyeli öğrenme metoduyla tasarlanan sistemlerde amaç ve ödüllendirmenin, etmenin başarısını etkilediği ve ne kadar önemli olduğu görülmektedir.

3.9.2. Değer fonksiyonu

Değer fonksiyonu, bir ödül fonksiyonunun uzun vadeli göstergesidir. Ödül fonksiyonu R, şimdiki durumun ve durum-eylem çiftinin anlık iyiliğini gösterir.

Buna karşılık değer fonksiyonu, mevcut durumdan başlayarak hedefe ulaşıncaya kadar alınan ödüllerin toplam değerini gösterir. Takviyeli öğrenmede değer fonksiyonu, bir hareket seçilirken gecikmiş ödülün göz önünde bulundurulmasını sağlar. Bir etmenin hareketi, sadece ortamdan alınan anlık ödüle değil, sonraki durumlara da bağlıdır. Sonraki durumun değeri etmen tarafından gecikmiş ödül olarak alınır. Etmenin uzun vadeli en iyilik modeli, gecikmiş ödülün öğrenme süresince nasıl kullanılacağını da açıklar. Gecikmiş ödülden öğrenen bir etmen bir çok hareketten sonra yüksek ödüllü bir duruma erişebilir[14].

3.9.3. Değer fonksiyonunun hesaplanması

Etmen, ortamındaki durumlar kümesi S'i algılayabilir ve yapabileceği A hareketlerinden birini uygulayabilir. Her bir ayrık zaman adımı t'de etmen, şu andaki durum s_t 'yi algılar ve ona uygun a_t hareketini yürütür. Ortam, etmene anlık ödül $r_t=r(s_t, a_t)$'ı vermekten ve bir sonraki durum $s_{t+1} = \delta(s_t, a_t)$'yi üretmekten sorumludur. Burada δ ve r fonksiyonları ortamın parçasıdır ve etmenin bunları bilmesine gerek yoktur. Bu algoritmada $\delta(s_t, a_t)$ ve $r(s_t, a_t)$ fonksiyonları sadece şu andaki durum ve harekete bağlıdır, daha öncekilere bağlı değildir. Burada S ve A'nın sonlu bir kümeye sahip olduğu göz önüne alınmıştır.

Etmenin amacı, şu anda gözlenen durum s_t 'ye göre bir sonraki a_t hareketini seçmek için $\Pi: S \rightarrow A$ politikasını, yani $\Pi(s_t)=a_t$ 'ı öğrenmektir. Π politikası etmenin zamanla en büyük toplam ödülü elde etmesiyle öğrenilebilir. Bunun için rasgele bir

başlangıç durumu, s_t'den rasgele bir politika takip edilerek elde edilen toplam ödül $V^{\Pi}(s_t)$ ile tanımlanabilir. Değer fonksiyonun hesaplanmasında kullanılan denklem aşağıda gösterilmektedir.

$$V^{\Pi}(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots$$
$$\equiv \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$

Denklem 3.2 Değer Fonksiyonun Hesaplanması

Burada ödüllerin sırası r_{t+i} , s_t durumundan başlanarak ve yukarıda tanımlandığı gibi, yani $a_t=\Pi(s_t)$, $a_{t+1}=\Pi(s_{t+1})$ ile hareketleri seçmede kullanılan Π politikası sayesinde üretilir. γ ise $0 \leq \gamma < 1$ arasında bir değer olup anlık ödüllere karşılık indirgenmiş ödül değerini belirler. Özellikle gelecek i zaman adımında alınan ödüller γ^i faktörüyle üstel bir şekilde azaltılır. Eğer $\gamma=0$ olursa, sadece anlık ödül göz önüne alınır. Eğer γ bire yaklaşırsa geleceğin ödüllerine anlık ödüle göre daha büyük önem verilir.

3.9.4. Takviyeli öğrenmenin amacı

Takviyeli öğrenmenin amacı optimal politikayı bulmaktır. Optimal politika, etmenin problemi optimal yönden çözmesini ve sonuca ulaşmasını sağlar. Böylece etmen hedefi olan en yüksek ödül değerine de ulaşmış olur. Optimal politika rasgele bir s_t durumundan başlanmış indirgenmiş toplam ödül miktarının maksimum değeri şeklinde ifade edilebilir.

$$\Pi = \arg\max(V^{\Pi}(s_t))$$

Denklem 3.3 Maksimize Edilmiş Toplam Ödül

Günümüze kadar birçok takviyeli öğrenme metodu önerilmiştir. Fakat q öğrenme algoritması bu algoritmalar arasında en yaygın olarak kullanılan algoritmadır[15].

3.10. Q Öğrenme

Q öğrenme popülaritesini her gün biraz daha arttıran bir takviyeli öğrenme algoritmasıdır. Q öğrenme [Watkins-1989] çevre hakkında bilgi sahibi olunamayan ortamlarda online olarak öğrenebilen bir algoritmadır. Q öğrenmenin amacı, rasgele bir ortam içinde etmene, optimal politikayı nasıl öğrenebileceğini göstermektir.

Etmenin optimal politikayı doğrudan öğrenmesi zordur. Çünkü etmen için kullanabileceği eğitim verileri bulunmamaktadır. Etmen için kullanılabilir eğitim verisi sadece anlık ödüllerdir. Bu eğitim verilerini kullanarak sayısal bir değerlendirme fonksiyonunu öğrenmek ve daha sonra bu fonksiyon yardımıyla optimal politikayı belirlemek daha kolaydır.

Etmen herhangi bir durumda bir eylem gerçekleştireceği zaman etmenin amacının toplam ödül miktarını maksimize etmek olduğunu söylemiştik. Buna göre optimal politika aşağıda verilen formülle ifade edilir:

$$\Pi^*(s) = \arg\max[r(s,a) + \gamma V^*(\delta(s,a))]$$

Denklem 3.4

δ(s, a), etmenin s durumunda a hareketi uyguladıktan sonra geldiği yeni durumu gösterir. Eğer etmen, anlık ödül fonksiyonu r(s,a) ve durum geçiş fonksiyonu δ(s,a) bilgisine sahip ise V^* 'yı öğrenerek optimal politikayı kazanabilir. Yani etmen,

hareketine cevap olarak ortam tarafından kullanılan r(s,a) ve δ(s,a) fonksiyonlarını bilirse, herhangi bir s durumu için optimal hareketi Denklem 3.4' ten hesaplayabilir.

Q algoritması için etmenin gerçekleştirebileceği her durum-eylem çifti Q(s,a) fonksiyonunu tanımlayalım. Q(s,a); s durumunda a hareketi yürütülerek anlık olarak alınan ödül ile bundan sonraki durumun optimal politikayı takip etme değerinin γ ile azaltılmış toplamıdır. Yani;

$$Q(s,a) \equiv r(s,a) + \gamma V^*(\delta(s,a))$$

Denklem 3.5

Q fonksiyonunu öğrenme, optimal politikayı öğrenmeye benzer. Burada anahtar problem, sadece anlık ödüller r' nin zamanla sırası verildikten sonra Q' nun eğitme değerlerini belirleyen yolu bulmaktır. Bu iteratif bir yaklaşımla başarılabilir. Bunun için aşağıda Q ve V^* arasında bir ilişki verilmiştir.

$$V^*(s) = \max_{a'} Q(s,a')$$

Denklem 3.6

Bu durumda Denklem 3.4'ü yeniden yazarsak;

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\delta(s,a),a')$$

Denklem 3.7

Q'nun bu recursive tanımı iteratif olarak Q'ya yaklaşan algoritmaların temelini oluşturur (Watkins, 1989). Öğrenici, gerçek Q fonksiyonunu tahmin edebilmesi için $\hat{Q}$ sembolünü kullanır. Bu sembol bir anlamda öğrenicinin Q hakkındaki hipotezidir. $\hat{Q}$,

her bir durum-hareket çifti için ayrı bir girişi olan tablo ile gösterilirse, her (s, a) çifti için tabloda $\hat{Q}(s,a)$ değeri vardır. $\hat{Q}(s,a)$ değeri öğrenicinin gerçek Q(s, a) değeri için şu andaki tahminini gösterir. Tablo başlangıç olarak rasgele değerlerle doldurulabilir. Fakat başlangıç değerleri olarak sıfır alınırsa algoritmayı anlamak daha kolay hale gelir. Q fonksiyonu için oluşturulmuş örnek bir tablo aşağıda gösterilmiştir.

	s1	s2	…	Sm
a1	Q(s1,a1)			
a2				
⋮			…	
An				Q(sm,an)

Tablo 3.2 Etmenin her bir durum-eylem çifti için hesapladığı Q değerlerini tuttuğu örnek tablo

Etmen şu andaki durum s'i sürekli olarak gözler, bir a hareketini seçer ve bu hareketi yürütür. Bu yapılırken ortaya çıkan ödül r=r(s, a) ve yeni durum $s^1 = \delta(s, a)$'i bir sonraki hareket için gözler. Daha sonra her bir geçiş için $\hat{Q}(s,a)$'yı aşağıdaki eğitme kuralına göre güncelleştirir.

$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a^1} \hat{Q}(s^1, a^1)$$

Denklem 3.8

Bu eğitme kuralı her ne kadar $\hat{Q}$ değerini kullansa da Denklem 3.7'den çıkarılmıştır. Denklem 3.7'de Q, $\delta(s, a)$ ve r(s, a)'ye göre tanımlansa da Denklem 3.8' deki eğitme kuralının bu fonksiyonları bilmesine gerek yoktur. Bunun yerine etmen, ortamındaki hareketi yürüttükten sonra meydana gelen yeni durum s^1 ve r ödülünü gözler ve daha sonra buna göre hareket eder.

3.10.1. Q Öğrenme Algoritması

Q öğrenme algoritması kısaca aşağıdaki gibi özetlenebilir:

- Her bir durum-eylem çifti için Q'(s,a) değerlerini sıfır veya sıfıra yakın değerlere ata
- Algılayıcılar vasıtasıyla etmenin içinde bulunduğu s durumunu gözle
- Öğrenme işlemi gerçekleşene kadar tekrarla
    - Bir a hareketi seç ve onu yürüt
    - Anlık ödül r'yi al
    - Yeni s^1 durumunu gözle
    - Q'(s,a) değerlerini aşağıdaki kurallara göre güncelleştir.

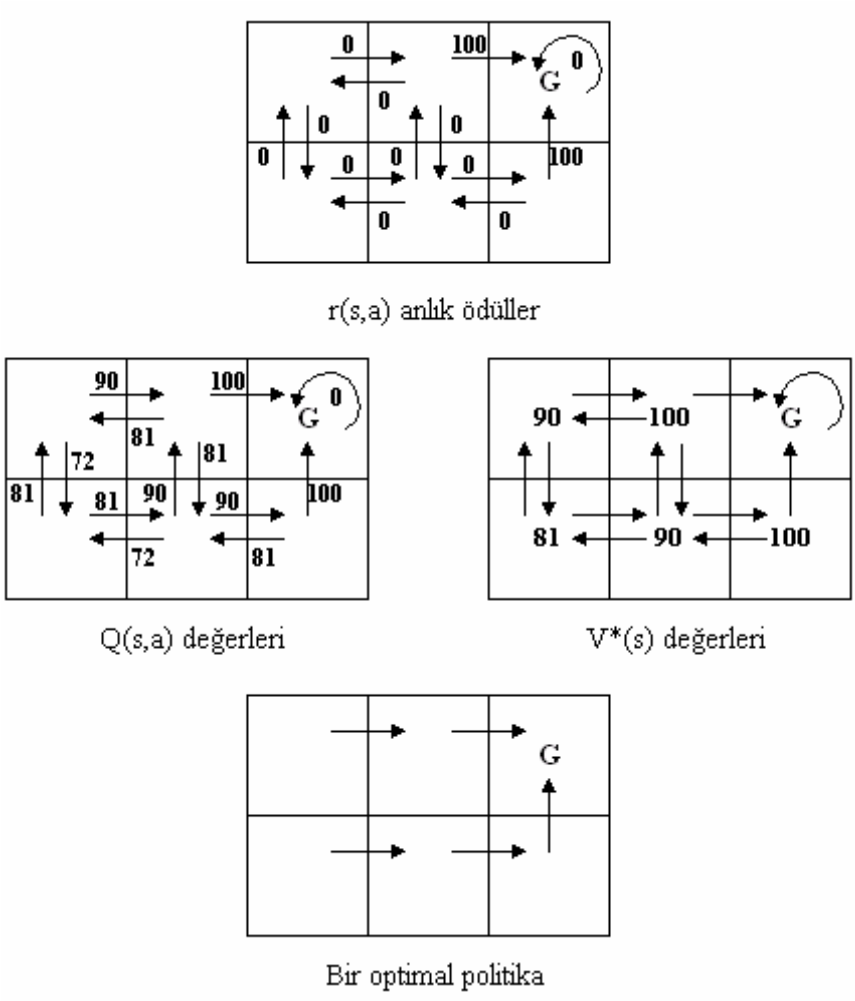
$$\hat{Q}(s,a) \leftarrow r + \gamma \max_{a^1} \hat{Q}(s^1, a^1)$$

    - $s^1 \rightarrow s$

3.10.2. Açıklayıcı bir örnek

Q öğrenmenin daha iyi açıklanması için aşağıda bir örnek verilmiştir. (6X6'lık bir *grid* üzerinde) En üstteki diyagramda 6 bölme, etmen için 6 mümkün durumu veya

yeri temsil eder. Diyagramdaki her bir ok etmenin bir durumdan diğerine geçebileceği mümkün hareketi temsil eder. Her bir okla ilişkilendirilmiş sayı, etmenin ilgili durum-hareket geçişini yürütmesi durumunda alacağı anlık ödül r(s, a)'yı gösterir. Bu özel ortamda anlık ödül, G durumuna geçme dışındaki bütün durum-hareket geçişleri için sıfırdır. G (*goal state*) amaç durumudur. Etmen G durumuna girdikten sonra yapabileceği tek hareket bu durumda kalmaktır. Bu nedenle G yutucu durum (*absorbing state*) olarak adlandırılabilir.(Watkins 1989)



Şekil-3.3: Ayrık durumlardan meydana gelen belirli bir etmen ortamı

Durumlar, hareketler, anlık ödüller tanımlandıktan ve azalma faktörü γ seçildikten sonra, optimal politika Π ve onun değer fonksiyonu $V^*(s)$ belirlenebilir. Örnek için γ=0.9 olarak alınmıştır. Şekil 3.3'ün en alttaki diyagramı bu ortam için bir optimal politikayı gösterir, fakat bundan başka optimal politikalar da vardır. Tüm politikalar gibi optimal politika da etmenin herhangi bir durumda seçeceği her hareketi belirleyebilir. Aynı zamanda optimal politika, etmeni G durumuna en kısa yoldan ulaştırır. Şeklin ortasındaki diyagram her bir durum için V^* değerlerini gösterir. Örneğin, bu diyagramda en alttaki sağ durum göz önüne alındığında bu durum için V^* değeri 100'dür. Çünkü bu durumda optimal politika, 100 anlık değeri olan yukarıya git hareketini seçer. Bundan sonra etmen yutucu durumda kalacak ve herhangi bir ödül almayacaktır.

Benzer şekilde en alttaki orta durum için V^* değeri 90'dır. Çünkü optimal politika, etmeni bu durumdan önce sağa (sıfır değerinde bir anlık ödül üreterek) sonra da yukarı doğru (100 anlık değeri üreterek) hareket ettirecektir. Böylece en alttaki orta durum için azaltılmış gelecek ödül;

$$0 + \gamma 100 + \gamma^2 0 + \gamma^3 0 + ... = 90 \text{ 'dır.}$$

Bu özel ortamda etmen yutucu durum G'ye ulaştıktan sonra onun sonsuz geleceği artık bu durumda kalmaktan ve sıfır ödüller almaktan ibarettir.

3.11. Takviyeli Öğrenmenin Geçmişi ve Yapılan Çalışmalara Bir Bakış

Takviyeli öğrenme mimarisinin geçmişi iki ana fikre dayanmaktadır. Bu fikirlerden bir tanesi hayvan öğrenme psikolojisinden faydalanarak bulunmuş deneme yanılma yoluyla öğrenme metodudur. Bu metot yapay zekanın ilk uygulamalarında kullanılmış ve 1980'lerin başında takviyeli öğrenmenin yeniden canlanmasını

sağlamıştır. Diğer fikir ise dinamik programlamayı ve değer fonksiyonlarını kullanarak bir problemin optimum çözüm yolunun bulunmasıdır[16].

Optimum kontrol terimi 1950'li yılların sonunda dinamik bir sistemin davranışlarını, sistem dışı zamanlarda minimize etmek için geliştirilen bir kontrol birimini tanımlamak için kullanılmıştır. Bu probleme bir yaklaşım da 1950'lerin ortalarında Richard Bellmann ve mesai arkadaşları tarafından getirilmiş, Hamilton ve Jacobi tarafından 19.yüzyılın teorisi olarak kabul edilmiştir. Bu kavram dinamik sistemlerde, değer fonksiyonlarında veya optimal dönüş fonksiyonlarında kullanılmaktadır ve günümüzde Bellmann eşitliği olarak da bilinmektedir[16].

Dinamik programlama, hedefe ulaşmak için en uygun imkanları seçmenin gerekli olduğu optimal kontrol problemlerinin çözümü için tek yol olarak bilinir. Dinamik programlamanın en önemli özelliği, değişen ortam koşullarına uyum sağlayarak verimliliğini devam ettirmesidir. Dinamik programlama son kırk yıl içinde geliştirilmiş, fonksiyon yaklaştırma ve asenkron metotlar gibi birçok uygulamada kullanılmıştır. Birçok mükemmel modern uygulamalarda dinamik programlamanın izleri bulunmaktadır (Örnek, Bertsekas, 1995; Puterman, 1994; Ross, 1983; and Whittle, 1982, 1983)[16].

Takviyeli öğrenme, optimal kontrol sistemlerini içeren problemlerde uygulanan bir öğrenme algoritmasıdır. Şimdi modern takviyeli öğrenmenin temelini oluşturan deneme yanılma ile öğrenme metodundan biraz bahsedelim. Deneme ve yanılma metodu hakkındaki ilk başarılı çalışma Edward Thorndike tarafından gerçekleştirilmiştir. Edward Thorndike gerçekleştirdiği eylemler sonrasında etmene belirli ödüllerin verilmesinden yazılarında bahsetmektedir. Edward Thorndike bu işleme etkinin yasası (Law of effect) diyerek; takviye işleminin, etmenin eylem seçmede önemli rol oynadığını söylemektedir[16].

Etkinin yasası, deneme yanılma metodu ile ilgili önemli iki bakış açısını içerir: Birincisi seçebilme özelliğidir. Metot gerçekleştirebileceği eylemler arasında seçim yapabilme özelliğine sahiptir. İkinci özelliği ise birliktelik özelliğidir. Metot ortam içerisinde diğer metotlarla birlikte çalışabilir. Denetleyerek öğrenme algoritmaları birliktelik özelliğine sahiptir fakat seçebilme yeteneğine sahip değillerdir. Bu iki metot takviyeli öğrenme algoritmasında birleşir. Deneme yanılma metodunun bir diğer avantajı da hafıza özelliğinin olmasıdır. Metot geçmişte gerçekleştirdiği eylemleri hafızasında tutar. Böylece aynı duruma tekrar geldiğinde hangi eylemin kendisi için daha iyi olduğunu kolaylıkla bilebilir[16].

Yapay zeka araştırmalarının başlarında deneme ve yanılma metodu birçok mühendislik uygulamasında kullanılmıştır. Minsky and Farley and Clark 1954 yılında birçok sayısal araştırmada bu metottan faydalanmıştır. Minsky takviyeli öğrenmenin sayısal modelleri üzerine birçok araştırma yapmış ve SNARC (Stochastic Neural-Analog Reinforcement Calculators) adını verdiği komponenti kullanarak analog bir makinenin yapısını oluşturmuştur. 1960'lı yıllarda takviye ve takviyeli öğrenme terimleri mühendislik literatüründe sıkça kullanılmaya başlanmıştır.(Örnek: Waltz and Fu, 1965; Mendel, 1966; Fu, 1970; Mendel and McClaren, 1970) Özellikle 1961 yılında Misnky' nin yayınladığı takviyeli öğrenme metoduyla kredi tayin etme konulu makale birçok uygulamaya örnek teşkil etmiştir[16].

Widrow, Gupta ve Maitra 1973 yılında, Widrow ve Hoff'un 1960 yılında geliştirdiği LMS algoritmasını geliştirerek, etmenin eğitim örnekleri dışında başarılı ve başarısız sinyalleri kullanarak öğrenmesini sağlamıştır. İkili, geliştirdikleri yeni yöntemle etmene blackjack oynamayı öğretmiştir[16].

1980'li yılların sonlarına gelindiğinde takviyeli öğrenme metotları daha çok labirent ve arama problemleri üzerinde uygulanmaya başlanmıştır ve 1989 yılında Watkins tarafından q öğrenme algoritması geliştirilmiştir. Q öğrenme algoritması günümüze kadar popülaritesini kaybetmeyen algoritmalardan bir tanesidir.

1990'lı yıllarda takviyeli öğrenme algoritmaları sanayi, robotik, hastalık teşhisi, v.b. birçok alana yayılmıştır. Günümüzde de elektronik alet ve robot tasarımlarında, fabrika otomasyonlarında ve bağımsız sistemlerde sıkça kullanılmaktadır.

Takviyeli öğrenmenin kullanıldığı bu geniş uygulama alanı metodun çalışmasına da belirli zorluklar getirmektedir. Takviyeli öğrenme metodundaki önemli bir mesele etmenin bilgiyi nasıl kazanacağı ve hafızasında nasıl tutacağıdır. Eğer öğrenme işleminin gerçekleştirileceği durum uzayı küçük ve ayrık ise q değerleri genellikle q tablosunda tutulur. Fakat bu yöntem daha geniş durum hareket uzaylarında pratik değildir ve sürekli durumlu uzaylarda mümkün değildir. Geniş durum uzaylı bir öğrenme işleminde sınırlı bir hafızada bütün durumları depolamak ve makul bir zamanda her bir durumu ziyaret etmek mümkün değildir. Bu gibi durumlarda çözüm; ziyaret edilen durumları, denetleyerek öğrenmede olduğu gibi ziyaret edilemeyenlerle genelleştirmektir. Bu problemlerin çözümlerinde fonksiyon yaklaşımlı yöntemler ve diğer yapay zeka algoritmalarıyla kullanılmış takviyeli öğrenme metotları kullanılmaktadır[14].

Fonksiyon yaklaşımlı yöntemler kullanılarak yapılan bir çalışmada etmen, ödül ve avcıların bulunduğu bir ortama konularak av-avcı uygulaması gerçekleştirilmiştir[13]. Etmenin geniş uygulama alanına uyum sağlaması için fonksiyon yaklaştırma yönteminden faydalanılmıştır.

Diğer bir çalışmada ise bulanık mantık ve takviyeli öğrenme teknikleri birleştirilerek geniş ve sürekli durum uzayı teknikleri ele alınır. Gerçekleştirilen uygulamada etmen, arama problemi içine konularak hedefine en kısa sürede ulaşması amaçlanmıştır. Etmenin yön seçme politikası için bulanık mantık kuralları ve mini-max algoritmasından faydalanılmıştır[2].

Son yıllarda geliştirilen yeni bir takviyeli öğrenme metodu da modüler takviyeli öğrenme algoritmasıdır. Bu yöntemde uygulama alanı etmenin öğrenme katsayısına göre bölümlere ayrılır. Böylece alt uygulama alanları elde edilir. Etmenin az öğrendiği alanlarda daha çok eylem gerçekleştirilirken daha çok öğrendiği alanlarda daha az eylem gerçekleştirilir. Böylece etmenin tüm uygulama alanını eşit olarak öğrenmesi amaçlanmıştır. Bu algoritmaya göre tasarlanan iki adet robota karşılıklı futbol oynatılması öğretilmiştir[17].

4. YAPAY SİNİR AĞLARININ TAKVİYELİ ÖĞRENMEYLE BİRLİKTE KULLANILMASI

Yapay sinir ağları (YSA) günümüzde bilgi sınıflama ve bilgi yorumlamanın içinde bulunduğu değişik problemlerin çözümünde kullanılmaktadır[18]. Yapay sinir ağları, çok farklı yapıda ve formlarda bulunabilen verileri hızlı bir şekilde tanıması ve algılamasından dolayı özellikle mühendislik alanında yapılan uygulamalara ışık tutmaktadır[19]. Ben de bu çalışmamda takviyeli öğrenme algoritması ile yapay sinir ağlarını birleştirerek yeni bir metot geliştirdim.

Yapay sinir ağları ile hızlandırılmış takviyeli öğrenme algoritması üzerine bilgi vermeden önce kullandığımız YSA teorisinden biraz bahsedelim.

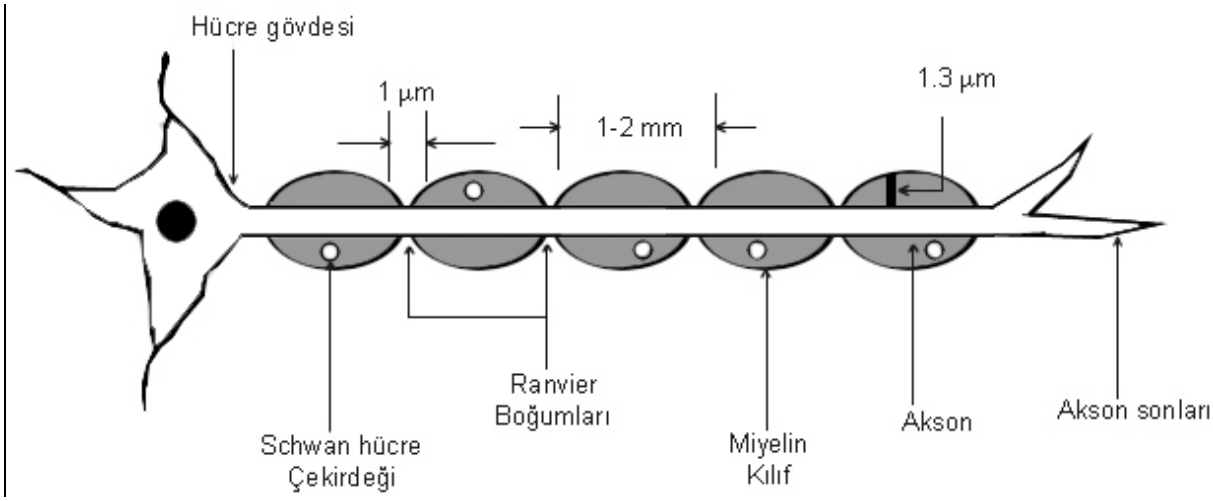
4.1. Kavram ve Terimler

YSA biyolojik sinir ağlarından esinlenerek geliştirilmiş bilgi işlem sistemleridir. Bir sinir ağı, bilgiyi depolamak için doğal eğilimi olan basit birimlerden oluşan paralel dağıtılmış bir işlemcidir. Bu ağlar, sinaptik ağırlıklar olarak bilinen nöronlar arası bağlantı kuvvetlerini, bilgiyi depolamak için kullanırlar. Benzer olarak YSA da birçok işlemciden oluşur ve farklı biçimlerde ifade edilebilen nümerik verileri taşıyan bağlantı yada ağırlıklarla birbirine bağlı yapılardır.

Biyolojik sinir ağlarında girdi sinyallerini alan, yorumlayan ve uygun çıktıyı ileten temel işlemci nöron olarak adlandırılır. Bir nöron, gövde (soma), gövdeye giren sinyal alıcıları (dentrit) ve gövdeden çıkan sinyal iletici (akson) olmak üzere üç kısımdan oluşur [20].

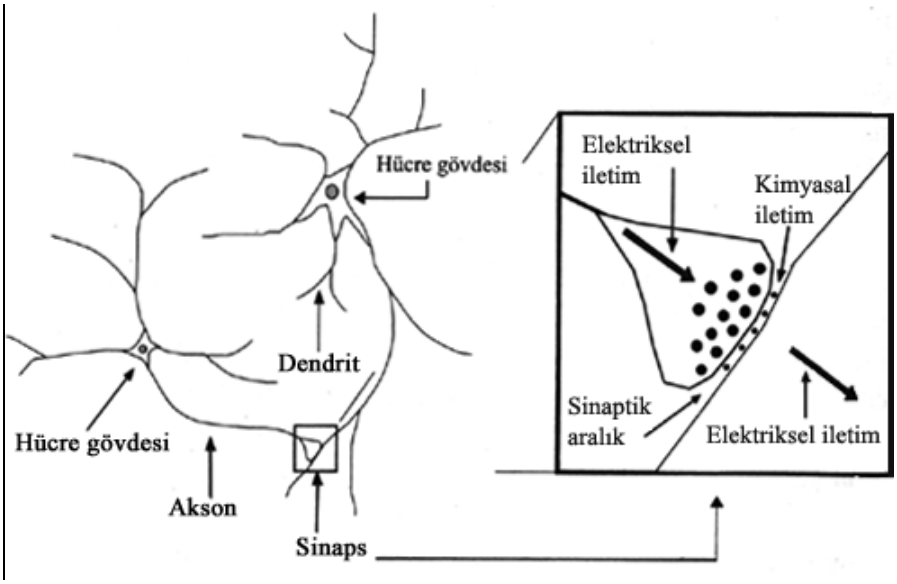
Dentritler, nörona bilgiyi alan ve sayısal olarak birden fazla olabilen yapılardır ve içyapıları nöronla aynıdır.

Aksonlar, dendritten aldığı bilgiyi diğer hücrelere aktaran uzantılardır. Uzunlukları birkaç mikrondan, 1-2 metreye kadar değişebilir. Her nöronun yalnızca bir aksonu vardır. Aksonlar akson kesecikleri denilen ve içlerinde çok miktarda vezikül bulunan bir çok yapı ile sonlanırlar. Bu keseciklerde sinaptik geçişte rol alan kimyasal taşıyıcılar (nörotransmitter) bulunur. Aksonlar özel bir örtüye sahip olmalarına göre miyelinli yada miyelinsiz olarak sınıflandırılabilirler. Akson üzerini örten miyelin kılıfın yalıtım ve darbe hızını arttırmak gibi iki önemli görevi vardır. Aşağıda miyelinli bir nöronun yapısı gösterilmiştir.



Şekil 4.3 Miyelinli bir nöron yapısı

Sinir hücreleri, dış kısmı pozitif, iç kısmı negatif yüklü ve ortasında çift lipid tabakasıyla yalıtılmış, kutuplanmış bir kondansatöre benzer. Bir uyarıcı ile bu kondansatör nötrlenebilir hatta anlık olarak ters kutuplanabilir. Zarın belirli bir bölgesinde 1ms kadar süren bu potansiyel değişiminden sonra zar bu bölgede eski haline dönerken, potansiyel değişikliği zar boyunca yayılmaya devam eder. Akson boyunca iletilen bilginin temel birim olan bu potansiyel değişimine aksiyon potansiyeli denir.

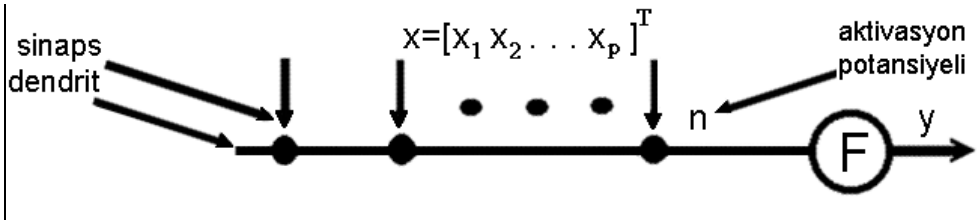


Şekil 4.2 Biyolojik Nöron

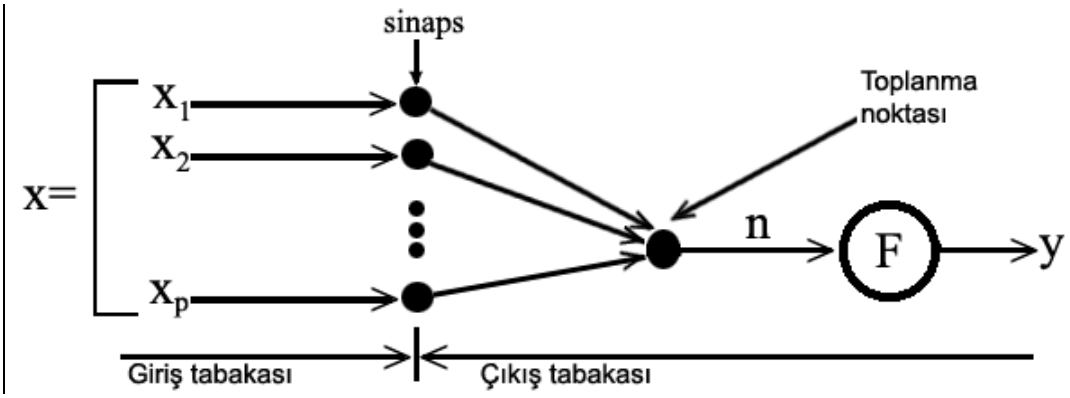
Sinir hücreleri arasında iletişimin gerçekleştiği, yapısal ve fonksiyonel olarak özelleşmiş bölgelere sinaps adı verilir. Mesajı gönderen ve presinaptik hücre olarak adlandırılan hücre ile mesajı alan ve postsinaptik hücre olarak adlandırılan hücre bu kavşak bölgelerinde birbirlerine oldukça yaklaşırlar.

Yapay sinir ağları, biyolojik sinir ağlarından esinlenerek, birbirine bağlı doğrusal ve/veya doğrusal olmayan birçok işlemci elemandan oluşur. Bir yapay nöron temel olarak girişler, ağırlıklar, toplam fonksiyonu, aktarım fonksiyonu ve çıkış olmak üzere beş kısımdan oluşur.

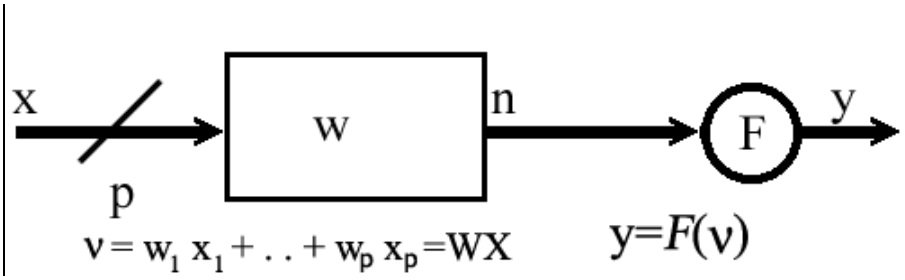
P girişli ve n çıkışlı bir yapay nöronun farklı grafik gösterimleri şöyledir:



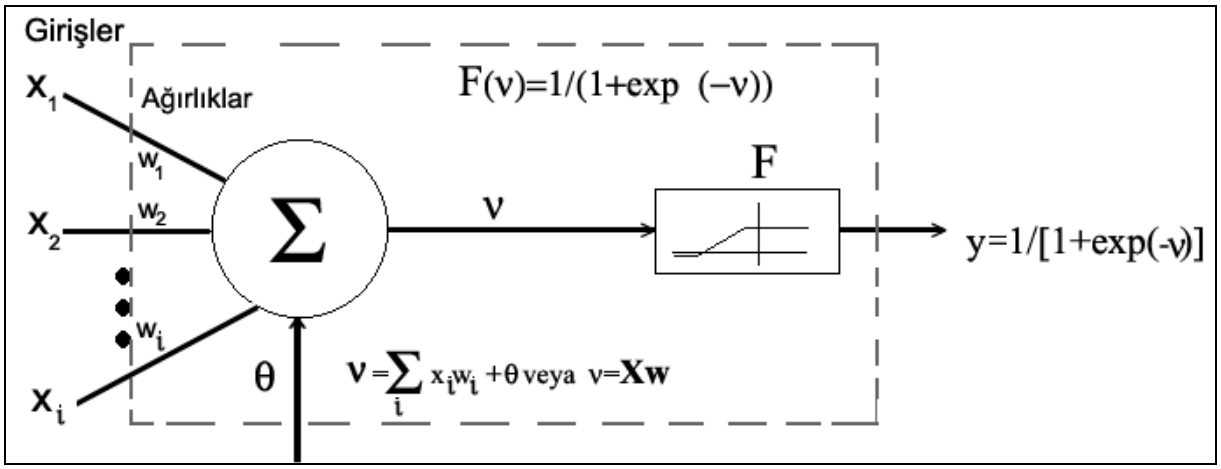
Şekil 4.3 Dendrit Gösterimi



Şekil 4.4 İşaret Akış Gösterimi



Şekil 4.5 Blok Diyagram



Şekil 4.6 Ayrıntılı Gösterim

Bu gösterimlerde **x** girişleri, **y** çıkışı, **F** aktivasyon fonksiyonunu, **p** nöron giriş sayısını, **n** çıkış sayısını, **w** ise ağırlıkları ifade eder.

Dendrit gösteriminde presinaptik aktiviteleri giriş işaretlerinin p elemanlı sütun vektörü olarak gösterilir. $x = [x_1, x_2, ..., x_p]^T$ giriş desenlerinin uzayı p boyutludur.

Sinapslar ağırlıklar olarak adlandırılan ayarlanabilir parametreler ile karakterize edilirler. Ağırlıklar, p elemanlı satır vektörü olarak düzenlenir:

$$W = \lfloor w_1, w_2, ..., w_p \rfloor$$

İşaret akış gösteriminde, p tane ağırlığı olan bir nöron giriş noktasının bir katmanı şeklinde düzenlenir. Ağırlıklar, giriş ile toplama noktası arasındaki bağlantılara karşılık gelir.

Sinapslardan ve dendritlerden geçen giriş işaretleri, "toplam post-sinaptik aktiviteyi tanımlayan aktivasyon potansiyeli" olarak toplanır.

Aktivasyon potansiyeli giriş işaretlerinin ve ağırlıklarının lineer toplamı olarak şekillenmiştir. Yani ağırlıklar ile geçiş vektörleri çarpımıdır:

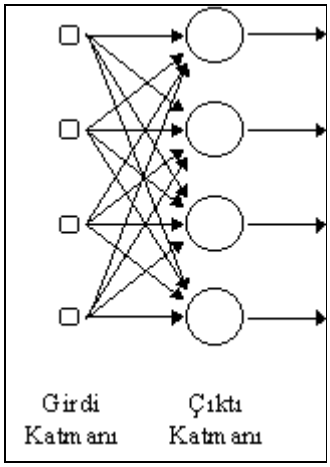
$$v = \sum_{i=1}^{p} w_i x_i = [w_1 w_2 ... w_p] \begin{bmatrix} x_1 \\ x_2 \\ \\ x_p \end{bmatrix} = WX^T$$

4.2. Yapay Sinir Ağlarının Sınıflandırılması

Yapay sinir ağlarında, ağ mimarisini belirleyen önemli etmenlerden biri de öğrenme algoritmasıdır. Genel olarak ağ mimarileri dört ana grupta toplanabilir [21]. Bu alt bölümde bu ağ mimarilerinin her biri hakkında kısa bilgi vereceğim.

4.2.1. Tek katmanlı ileri beslemeli ağlar

Katmanlı modellerdeki en basit ağ tipi olup bir çıktı katmanı ve buna bağlı bir girdi katmanı bulunmaktadır. Aşağıdaki şekilde tek katmanlı ileri beslemeli bir ağ örneği gösterilmektedir.

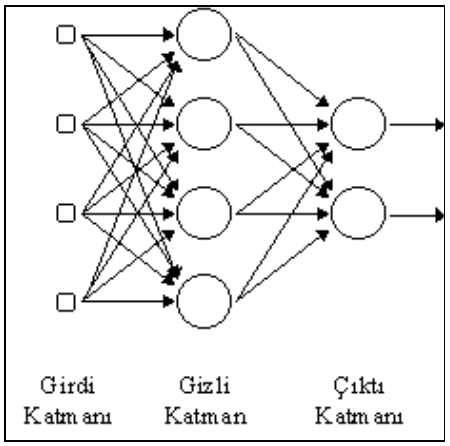


Şekil 4.7 Tek Katmanlı İleri Beslemeli Ağ Modeli

4.2.2. Çok katmanlı ileri beslemeli ağlar

Tek katmanlı ağlardaki girdi ve çıktı katmanından başka, bir yada daha fazla sayıda gizli katman içeren ağlara çok katmanlı ağ denir. Dış dünya tarafından doğrudan

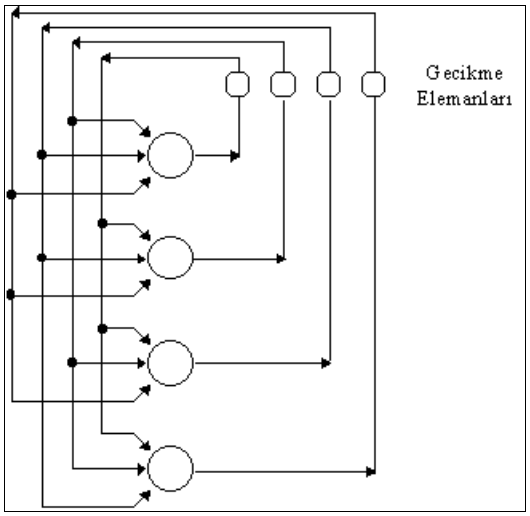
müdahale edilmediği için gizli katman adı verilen katmanda bulunan birimlere de gizli birimler adı verilir. Aşağıdaki şekilde çok katmanlı ileri beslemeli bir ağ örneği gösterilmektedir.



Şekil 4.8 Çok Katmanlı İleri Beslemeli Ağ Modeli

4.2.3. Geri dönümlü ağlar

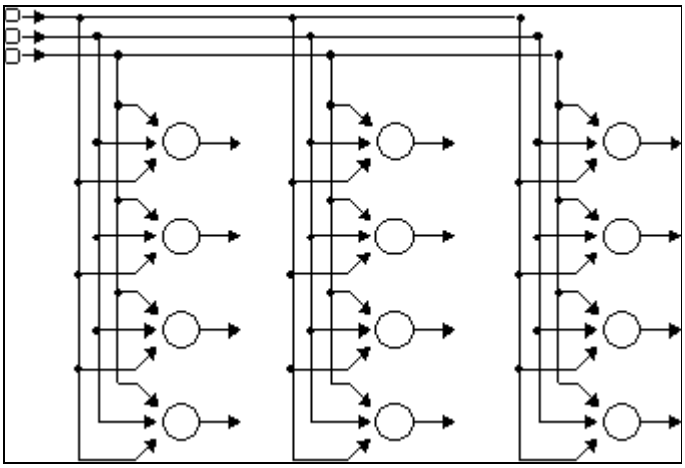
Geri dönümlü ağların ileri beslemeli ağlardan farkı, bu ağlarda en az bir tane geri-besleme döngüsünün bulunmasıdır.



Şekil 4.9 Geri Dönümlü Ağlar

4.2.4. Kafes yapılı ağlar

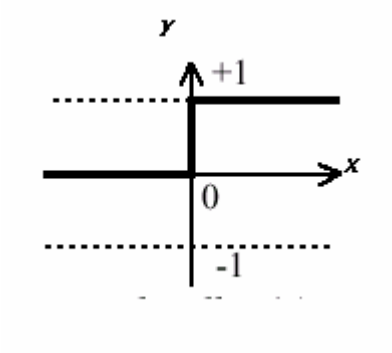
Kafes yapılı ağlar bir, iki veya çok boyutlu nöron dizinlerinden meydana gelmekte olup girdi katmanı, dizinlerin hepsine girdi bilgisi sağlamaktadır. Aşağıdaki şekilde kafes yapılı bir ağ örneği gösterilmektedir.



Şekil 4.10 Kafes Yapılı Ağlar

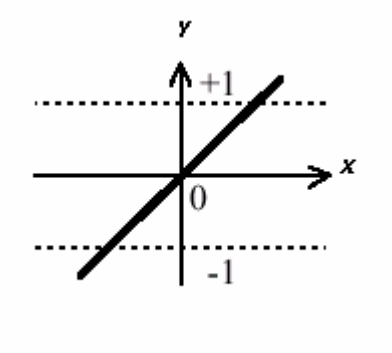
4.3. Aktivasyon Fonksiyonları

Aşağıdaki şekilde eşik aktivasyon fonksiyonunun grafiği görülmektedir. Eşik aktivasyon fonksiyonu eğer net değeri sıfırdan küçükse sıfır, sıfırdan daha büyük bir değer ise net çıkışında +1 değeri verir. Eşik aktivasyon fonksiyonunun –1 ile +1 arasında değişeni ise signum aktivasyon fonksiyonu olarak adlandırılır. Signum aktivasyon fonksiyonu, net giriş değeri sıfırdan büyükse +1, sıfırdan küçükse –1, sıfıra eşitse sıfır değerini verir.



Şekil 4.11 Eşik Aktivasyon Fonksiyonu

Aşağıdaki şekilde doğrusal aktivasyon fonksiyonu görülmektedir. Lineer aktivasyon fonksiyonunun çıkışı girişine eşittir. Sürekli çıkışlar gerektiği zaman çıkış katmanındaki aktivasyon fonksiyonunun lineer aktivasyon fonksiyonu olabildiğine dikkat edilmelidir.

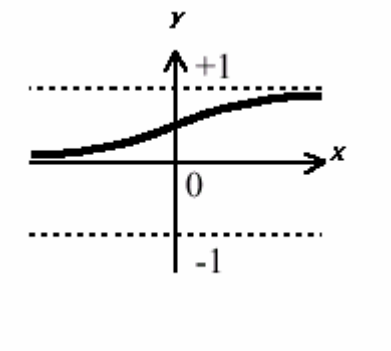


Şekil 4.12 Doğrusal Aktivasyon Fonksiyonu

$$f(x) = x$$

Denklem 4.1

Aşağıdaki şekilde sigmoid transfer fonksiyonu görülmektedir. Lojistik fonksiyon olarak da adlandırılmaktadır. Bu fonksiyonunun lineer olmamasından dolayı türevi alınabilmektedir. Böylece sigmoid transfer fonksiyonunu geri beslemeli ağlarda kullanmak mümkün olabilmektedir.



Şekil 4.13 Doğrusal Aktivasyon Fonksiyonu

$$f(x) = \text{lojistik}(x) = \frac{1}{1+\exp(-\beta x)}$$

Denklem 4.2

Buradaki β eğim sabiti olup genelde bir olarak seçilmektedir. Diğer bir aktivasyon fonksiyonu olan hiperbolik tanjant aktivasyon fonksiyonu da lineer olmayan türevi alınabilir bir fonksiyondur. +1 ile –1 arasında çıkış değerleri üreten bu fonksiyon lojistik fonksiyona benzemektedir. Denklemi aşağıda görüldüğü gibidir.

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Denklem 4.3

Bu aktivasyon fonksiyonlarından başka fonksiyonlar da vardır. Yapay sinir ağında hangi aktivasyon fonksiyonunun kullanılacağı probleme bağlı olarak değişmektedir. Yukarıda verilen fonksiyonlar en genel aktivasyon fonksiyonlarıdır.

4.4. Yapay Sinir Ağlarında Öğrenme

Yapay sinir ağlarında bilgi, nöronlar arasındaki bağlantılar üzerindeki ağırlık değerleri üzerinde tutulur. Bu yüzden yapay sinir ağlarının eğitimi ve öğrenme bu ağırlıklara değerlerin verilmesi ve değiştirilmesi anlamına gelmektedir. Öğrenme yaklaşımı, ağ yapıları için ayırt edici bir özelliktir. En genel anlamda öğrenme, güdümlü öğrenme ve güdümsüz öğrenme olmak üzere iki grupta toplanabilir[21].

4.4.1. Güdümlü Öğrenme

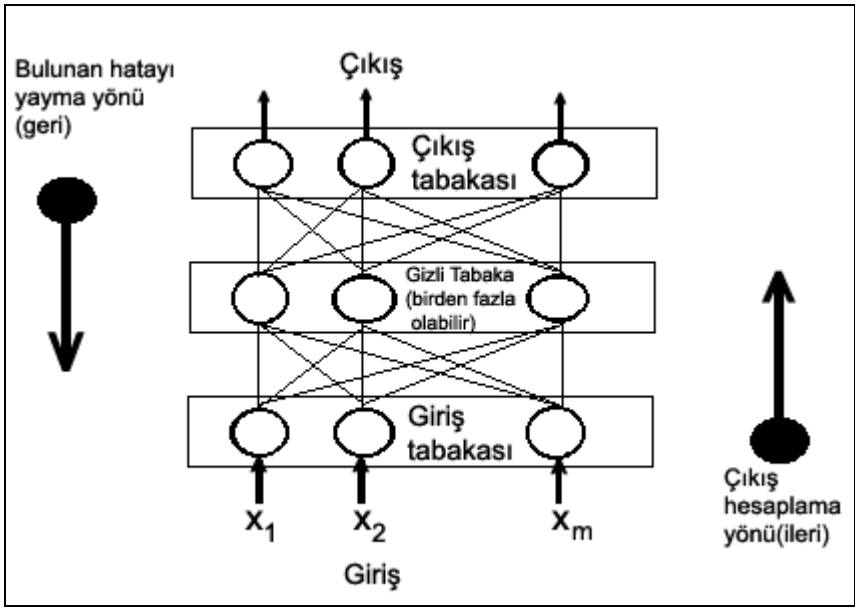
Güdümlü öğrenme, yapay sinir ağlarında yaygın olarak kullanılan bir yöntemdir. Eğitim sırasında sisteme bir girdi ve hedef çıktı vektörü çifti verilir. Giriş bilgisinden üretilen çıktı değeri ile hedef çıktı değeri kıyaslanarak ağırlıkların değiştirilmesi için gerekli bilgi hesaplanır. Bu kıyaslamadan elde edilen fark, hata olarak belirlenen değerden küçük oluncaya kadar eğitim devam eder. Hata değeri istenen değerin altına düştüğünde ağırlık değerleri sabitlenir ve eğitim biter.

4.4.2. Güdümsüz Öğrenme

Kullanılan girdi vektörlerinin hangi sınıfa ait olduğu verilmiyorsa, bu sınıflama işlemini girdileri birbirleriyle karşılaştırarak sistemin kendisi yerine getiriyorsa, bu tür sistemlerdeki öğrenme yöntemine *güdümsüz öğrenme* adı verilir. Bu sistemlerde bir grup girdi vektörü sisteme verilir, ancak hedef çıktılar belirtilmez. Sistem girdiler içinde birbirine en çok benzeyenleri gruplar ve her bir grup için farklı bir örüntü tanımlar. Öz düzenlemeli özellik haritaları bu yöntemi kullanarak sınıflama işlemini yerine getirir.

4.5. Çok Katmanlı Perseptron ve Geri Yayılımlı Öğrenme

Bu bölümde geliştirdiğim metotta faydalandığım güdümlü öğrenme metotlarından Çok Katmanlı Perseptron (Multi Layer Perceptron-MLP) ve Geri Yayılımlı Öğrenme (Back propagation) hakkında bilgi vereceğim. Çok katmanlı perseptronlar; ileri beslemeli, tam bağlantılı, bir girdi katmanı, bir veya birden fazla gizli katman ve bir çıktı katmanından oluşan hiyerarşik yapıya sahip yapay sinir ağı mimarileridir. Böyle bir ağdaki herhangi bir nöronun çıkışı $y_k = f\left(\sum w_k x\right)$ ile hesaplanır. Giriş katında herhangi bir bilgi işleme yapılmaz. Buradaki işlem elemanı sayısı problemin giriş sayısına bağlıdır. Ara katman sayısı deneme yanılma yoluyla bulunur. Çıkış katmanı eleman sayısı ise uygulanan probleme bağlıdır[21].



Şekil 4.14 Çok Katmanlı Perseptron Yapısı

Geri yayılımlı öğrenme metodu olarak isimlendirilen eğitim algoritmasının temeli ilk olarak Werbos'un Harvard Üniversitesi'nde verdiği doktora tezinde ileri sürülmüş (Werbos, 1974), daha sonra Parker tarafından MIT'nin bir teknik raporunda

ele alınmış (Parker, 1985) ve Rumelhart ve arkadaşları tarafından da popüler hale getirilip uygulanabilir şekle dönüştürülmüştür.

Daha önce Adaline mimarisi ile kullanılan LMS veya Delta kuralının geliştirilmiş ve çok katmanlı yapıya uyarlanmış hali olan geri yayılımlı öğrenme metodu, çok katmanlı perseptron yapısı üzerinde çok çeşitli problemlerin çözümünde kullanılmaktadır. Burada geri yayılım kavramı, oluşan hatalar için kullanılmaktadır.

Geri yayılımlı öğrenme metodu ile çok katmanlı perseptron ağı üzerinden iki geçiş söz konusudur. İleri besleme aşamasında eğitim için kullanılan girdi sisteme beslenir ve bunun sonuçları her bir katmanı geçerek çıktı katmanına kadar gelir ve girdiye karşılık bir çıktı elde edilir. Bu aşama sırasında ağ üzerindeki ağırlık değerleri sabit tutulur. İkinci aşama olan geri besleme aşamasında, elde edilen çıktı ile hedef çıktı arasındaki farktan hata sinyali elde edilir ve bu sinyal ağ yapısında geriye doğru yayılır. Bu geri yayılım aşamasında, oluşan hatayı minimuma indirecek şekilde ağırlık değerleri güncellenir.

Bu güdümlü öğrenme işleminin tekrarlanması ile sistem, kendisine beslenen girdi örneklerini öğrenir. Eğitim bittikten sonra sistemin uygulama sürecinde sadece ileri beslemeli ilk aşama yeterli olmaktadır. Bu durumda sistemin eğitilmesi uzun süre almakla beraber, eğitilmiş bir sistemden bilgi alınması çok hızlı olmaktadır.

4.6. Takviyeli Öğrenmede Yapay Sinir Ağları

Çevre hakkında bilgi sahibi olunamayan ortamlarda öğrenme ve çevreye uyum sağlama büyük önem kazanmaktadır. Eğer etmenler yaşadıkları çevrenin dinamikleri hakkında çok az bilgiye sahiplerse takviyeli öğrenme teknikleri kullanılabilir. Bu çalışmamda engellerle karmaşık hale getirilmiş basit bir labirent ortamında yapay sinir ağlarıyla hızlandırılmış takviyeli öğrenme teknikleri üzerinde durdum. Labirent ortamına

konulan etmen, çıkışı bulmak için öğrenmek zorundadır. Etmen için amaç çıkışı en kısa yoldan bulabilmektir. Geliştirilen labirent ortamı çok büyük durum uzayına sahiptir. Etmenin bu ortam içerisinde q öğrenme tekniklerini kullanarak çıkışı bulması çok zaman almaktadır. Ben de etmenin daha hızlı hedefine ulaşabilmesi için q öğrenme algoritmasının yanında yapay sinir ağlarından faydalandım.

Bu çalışmada yapay sinir ağları ile hızlandırılmış takviyeli öğrenme metodu kullanılarak labirent ve arama problemlerine daha hızlı çözüm bulma ileri sürülmüştür. Geliştirilen yeni metottan, deneysel sonuçlardan ve önerilerden aşağıda bahsedilmektedir.

5. YAPAY SİNİR AĞIYLA HIZLANDIRILMIŞ TAKVİYELİ ÖĞRENME

Geliştirilen metot; etmen ve çevrenin özellikleri, yapay sinir ağlarıyla hızlandırılmış takviyeli öğrenme mimarisi, hızlandırılmış takviyeli öğrenme algoritması, bilgisayar simülasyonu ve deneysel sonuçlar olmak üzere dört bölümde anlatılacaktır.

5.1. Etmen ve Çevrenin Özellikleri

Takviyeli öğrenme küçük çocuklardaki gibi yapılan hatalardan ders çıkarılması üzerine kurulmuştur. Bir çevrede görevlendirilmiş bir etmen amacını bilir ve içinde bulunduğu durumu amacını da içeren bir duruma nasıl çevirmesi gerektiğini öğrenir. Eğer etmen bunu yapabilmek için minimum insan yardımı alıyorsa bu etmene bağımsız etmen denir. Bu etmenler çevreleriyle ilişkiye girerek takviye sinyali alırlar ve bu sayede öğrenirler. Çevreden bu şekilde öğrenmek iyi bir metottur. Çünkü bu şekilde etmen direk çevresinde bulunan dinamiklerle etkileşim içerisindedir.

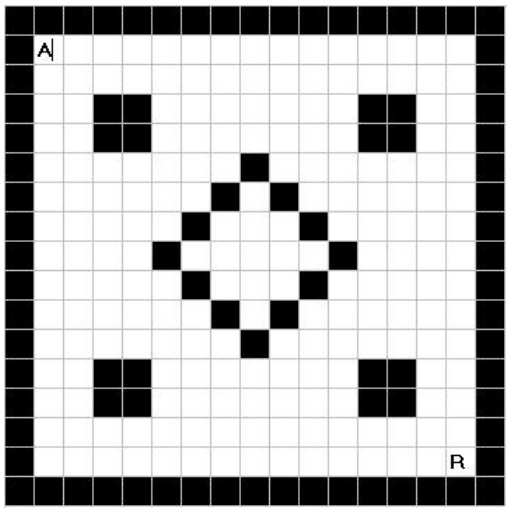
Etmen içinde bulunduğu çevreyi ne kadar iyi algılayabilirse ve çevre içinde ne kadar esnek hareket edebilirse öğrenme işlemi o kadar hızlı gerçekleşir. Etmenin içinde bulunduğu çevrenin karmaşıklığı çevrenin özellikleriyle, etmenin hareketliliği ise etmenin özellikleriyle ilgilidir.

5.1.1. Çevrenin özellikleri

Bu çalışmamda 15*15'lik bir yaşamsal ızgara ortamı kullandım. Kullandığım bu ızgara ortamında çeşitli varlıklar bulunmaktadır. Bunlar:

- Etmen
- Engeller
- Ödül

Yapılan çalışmada etmen, çevresindeki varlıklar hakkında bilgi edinerek ödüle giden en optimum yolu öğrenmeye çalışmaktadır. Engellerin görevi ortamın tekdüze olmamasını sağlamak ve etmenin amacına ulaşmasını zorlaştırmaktır. Etmen hareket halindeki varlık iken ödül ve engeller hareketsiz varlıklardır. Bu çalışmada etmen ve ödül sayısı bire sabitlenirken engel sayısı ortamın karmaşıklığına göre değişebilmektedir. Çok karmaşık ortamlarda engel sayısı fazla iken az karmaşık ortamlarda engel sayısı azdır. Geliştirilen çevreye ait bir örnek aşağıdaki şekilde gösterilmektedir.



Şekil 5.1 Tek etmen ve tek ödüllü çevre

Geliştirilen metotta kullanılan çevre örneğine göre çevrede üç çeşit varlık bulunmaktadır. Etmen A harfi ile, ödül R harfi ile ve engeller de koyu dikdörtgenler şeklinde gösterilmiştir. Sonuç olarak geliştirilen ortam, takviyeli öğrenme kategorisine girer. Ortam q öğrenme algoritması için özel bir durum olarak düşünülebilir.

5.1.2. Etmenin özellikleri

Takviyeli öğrenmede öğrenen etmenler, çevreleriyle iletişim kurabilmek için algılama, insan gibi öğrenme ve etkili yeteneklere ihtiyaç duyarlar. Etmen modelleme, etmenin çevre içindeki performansını direk olarak etkilemektedir. Geliştirilen metotta etmen çeşitli algılayıcılara sahiptir. Böylece etmen çevre hakkında bilgi toplayabilecek ve değişen çevre durumlarını gözlemleyebilecektir. Etmen için geliştirilmiş iki çeşit algılayıcı vardır. Bunlar:

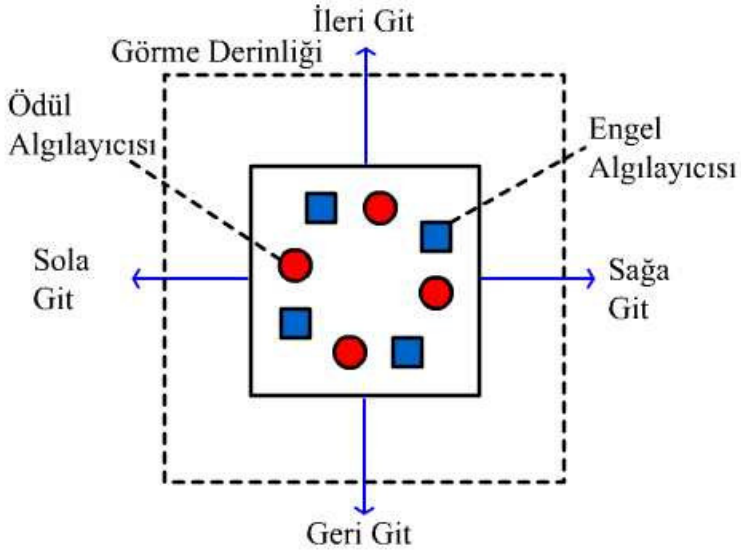
- Ödül algılayıcısı
- Engel algılayıcısı

Etmen üzerindeki algılayıcılar sadece belirli bir varlığı algılayabilir. Bu sebepten etmen üzerine yerleştirilen algılayıcılar sadece belirli noktalarda belirli varlıklar bulunması durumunda aktif hale gelir. Etmen üzerinde sekiz adet algılayıcı bulunmaktadır. Bunların dört tanesi ödül algılayıcısı, dört tanesi ise engel algılayıcısıdır. Etmen kare şekilde tasarlanmış olup etmenin her kenarına bir tane ödül, bir tane de engel algılayıcısı konulmuştur. Etmen, algılayıcılarından elde ettiği bilgilere göre çevre içinde hareket eder. Etmenin gerçekleştirebileceği hareketler aşağıdaki gibidir:

- Sağa gitmek
- Sola gitmek
- Yukarı gitmek
- Aşağı gitmek

Etmen, çevre içinde gezinirken bütün durumlarda bütün hareketleri gerçekleştiremeyebilir. Etmenin çevre dışına ve engellerin bulunduğu karelere yaptığı hareketlerin hepsi illegaldir. Ortamdaki her karede zamanın herhangi bir adımında sadece bir varlık bulunabilir. Etmen eğer ödül bulunan kareye gelirse çevre tarafından

ödül etmene verilir. Etmen ve ödül başlangıç pozisyonlarına konumlandırılarak öğrenme işlemi devam ettirilir. Öğrenme işleminin optimum olabilmesi için etmenin çevre içindeki illegal hareketleri engellenmelidir. İllegal hareketlerin etmen tarafından algılanması ve bu hareketlerin seçilmesinin engellenmesi için yapay sinir ağları kullanılmaktadır. Geliştirilen metotta yapay sinir ağlarından iki aşamada faydalanılmaktadır. Birinci aşamada yapay sinir ağları, etmenin herhangi bir durumda yapabileceği hareketleri belirlerken, ikinci aşamada etmenin yön seçmesine yardımcı olarak, etmenin daha hızlı bir şekilde optimum yolu bulmasını sağlamaktadır. Geliştirilen etmenin özelliklerini gösteren resim aşağıda yer almaktadır.



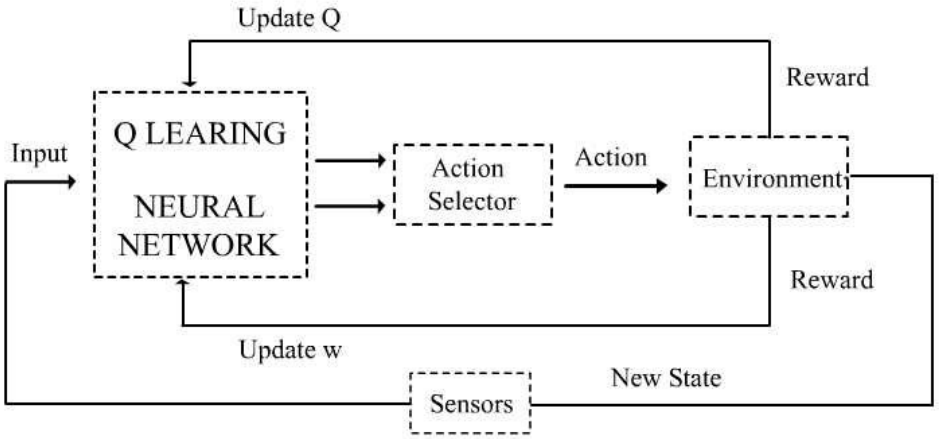
Şekil 5.2 Geliştirilen Etmen

Belirlenen özellikler ve kurallara göre etmenin yapması gerekenleri şöyle sıralayabiliriz.

- Ödülü bulma
- Engele çarpmama
- Eğer görünürde ödül yoksa ödülü arama

5.2. Sinir Ağıyla Hızlandırılmış Takviyeli Öğrenme Mimarisi

Labirent ve arama problemlerinin çözümü için ilk akla gelen algoritma q öğrenmedir. Sinir ağları ile hızlandırılmış takviyeli öğrenme algoritması, sinir ağlarının bilgi yorumlama özelliğinden faydalanılarak geliştirilen bir takviyeli öğrenme metodudur. Labirent ve arama problemlerinde ortamın büyütülmesi ve engellerle daha karmaşık hale getirilmesi q öğrenmenin işini zorlaştırmaktadır. Sayısı katlanarak artan q değerleri ve büyüyen q tablosunun güncelleştirilmesi oldukça zaman almaktadır. Ben bu gibi durumlara çözüm bulmak için bu metodu geliştirdim. Sinir ağlarıyla hızlandırılmış takviyeli öğrenme, etmenin çevre içinde hareket ederken daha yüksek q değerlere sahip durumlara daha çabuk gelmesini sağlamaktadır. Böylece etmen amacına giden yoldaki q değerlerini daha çabuk güncelleştirerek optimum yolu en kısa zamanda bulabilmektedir. Yapay sinir ağlarıyla geliştirilen takviyeli öğrenme metoduna ait yapı Şekil5.3' te gösterilmektedir.

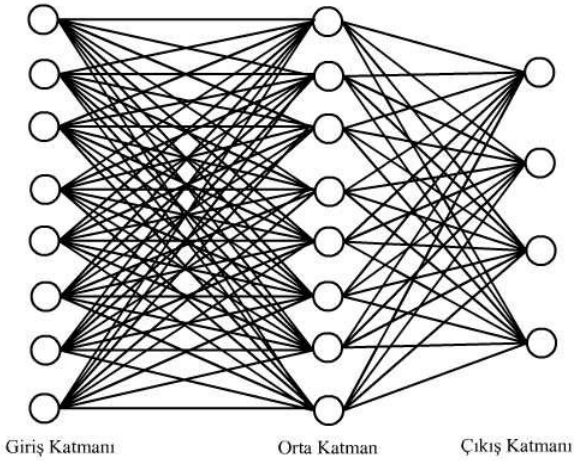


Şekil 5.3 Yapay sinir ağlarıyla hızlandırılmış takviyeli öğrenme mimarisi

Etmen ilk başta algılayıcıları vasıtasıyla çevresinden gelen durumları algılamaktadır. Algıladığı durumları yapay sinir ağına bildirerek gerçekleştirebileceği eylem kümesini oluşturmaktadır. Daha sonra q öğrenme ve yapay sinir ağlarını beraber

kullanarak eylem seçme işlemini gerçekleştirmektedir. Etmen, eylemi gerçekleştirdikten sonra çevreden aldığı ödül değerini ve q öğrenme algoritmasını kullanarak q değerlerini ve seçtiği eyleme ait ağırlık değerlerini güncellemektedir. Etmen optimum yolu bulup öğrenme işlemi sona erene kadar yukarıdaki işlemlere devam etmektedir.

Geliştirilen metotta giriş,orta ve çıkış olmak üzere üç katmandan oluşan bir yapay sinir ağı kullanılmıştır. Kullanılan yapay sinir ağının yapısı Şekil 5.4' te gösterilmektedir.

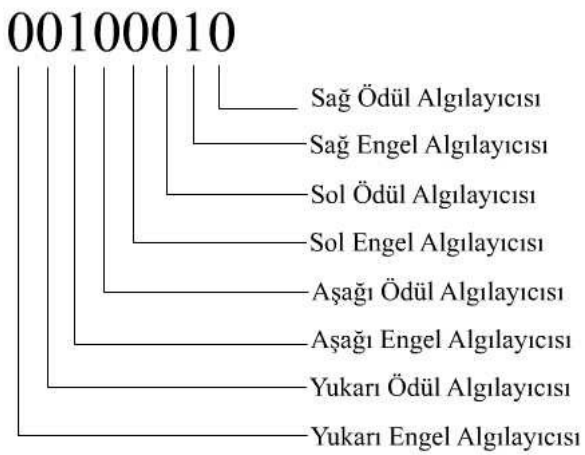


Şekil 5.4 Kullanılan yapay sinir ağı modeli

Yapay sinir ağı etmenin özelliklerine göre geliştirilmiştir. Etmenin algılayıcıları tarafından alınan değerler giriş katmanını oluşturmaktadır. Giriş katmanında sekiz adet nöron bulunmaktadır. Etmenin yapabileceği hareketler ise yapay sinir ağının çıkış katmanını oluşturmaktadır. Çıkış katmanında dört adet nöron bulunmaktadır. Orta katmanda ise ben sekiz adet nöron kulandım. Orta katmanda kullanılacak nöron sayısı programcı veya kullanıcı tarafından değiştirilebilir. Fakat giriş ve çıkış katmanlarındaki nöron sayılarında değişiklik yapılamaz.

Yapay sinir ağından, birinci aşamada, etmenin herhangi bir durumda yapabileceği eylemler kümesini belirlemek için yararlanılmaktadır. Bunun için yapay

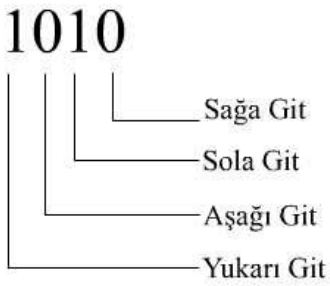
sinir ağımız simülasyon programının başlangıcında rasgele ağırlık değerleriyle ayarlandıktan sonra daha önceden belirlenmiş eğitim örneklerine göre eğitilmektedir. Bu eğitim örneklerine göre eğitilmiş yapay sinir ağı, etmenin engelleri algılamasını ve herhangi bir durumda illegal hareket yapmasını engellemektedir. Böylece etmenin ortam dışına çıkmaması, engel olan karelere ilerlememesi ve herhangi bir ödül gördüğünde alması sağlanmaktadır. Yapay sinir ağının giriş katmanını oluşturan sekiz bitlik eğitim örneği aşağıdaki resimde gösterilmektedir.



Şekil 5.5 Giriş katmanı için sekiz bitlik eğitim örneği

Etmenin algılayıcıları tarafından algılanan durumlar “1” ile, algılanmayan durumlar ise “0” ile gösterilmektedir. Etmenin herhangi bir yönde hem etmeni hem de engeli algılaması gibi bir durum söz konusu değildir. O yüzden aynı yöne bakan algılayıcılardan aynı zaman adımında sadece bir tanesi aktif konumda bulunabilir.

Simülasyon programında etmenin görme derinliği bir kare olarak ayarlanmıştır. Eğitim aşamasında etmene, verilen giriş eğitim örneklerine karşılık çıkış eğitim örnekleri de verilir. Yukarıda verilen giriş eğitim verilerinin sonucu olan çıkış eğitim verileri aşağıda gösterilmektedir:



Şekil 5.6 Çıkış katmanı için dört bitlik eğitim örneği

Etmenin bulunduğu durumda yapabileceği hareketler “1” olarak, yapamayacağı hareketler ise “0” olarak belirlenmiştir. Yapay sinir ağı eğitimi için yukarıdaki formatta yirmi adet eğitim örneği belirlenmiştir. Bu eğitim örnekleri etmenin engel ve ödüllerle karşılaşabileceği bütün durumları içermektedir. Yapay sinir ağı eğitimi için hata payı 0.00001 değerinde alınmıştır. Eğer yapay sinir ağı bu hata payı değerinden daha az değerde hataya ulaşırsa ağ eğitimi otomatik sonlandırılmakta ve etmen çevre içinde hareket etmeye başlamaktadır.

İkinci aşamada yapay sinir ağından, gerçekleştirilebilecek hareketler arasında eylem seçme işlemi için faydalanılmaktadır. Bu aşamada çıkışa bağlı ağırlık değerleri, q algoritması ve çevre tarafından verilen ödül değerleri yardımıyla güncelleştirilmektedir. Bu güncelleştirme işlemi etmenin ödül aldığı yönlere ilerlemesini ve q değerleri yüksek olan karelere daha çabuk gelmesini sağlamaktadır. Böylece q öğrenme algoritması hızlanarak amacı olan optimum yolu bulma işlemini daha az denemede ve daha kısa zaman içerisinde öğrenebilmektedir.

Herhangi bir durum için q algoritması tarafından elde edilen değerler Q(s,a), yapay sinir ağı çıkışları tarafından elde edilen değerler ise $Q_{nn}(s,a)$ şeklinde tanımlanmaktadır. Bu iki değer de eylem seçme işleminde etmen tarafından dikkate alınmaktadır.

5.2.1. Eylem seçme

Dikkat edilirse q algoritması, hareketlerin etmen tarafından nasıl seçileceğini belirlemez. Etmen eylem seçmede aç gözlü bir politika izler. Öncelikle etmen güncel durumdaki gerçekleştirebileceği eylemleri belirler. Daha sonra sinir ağından gelen $Q_{nn}(s,a)$ değerleri ile q öğrenme algoritmasından gelen Q(s,a) geçiş değerlerini beraber hesaba katarak kendisine maksimum değeri veren yönü seçer. Aşağıda etmenin yön seçme işlemi için kullandığı denklem gösterilmektedir.

$$s(a) = \arg\max(Q_{nn}(s,a) + Q(s,a))$$

Denklem 5.1 Eylem Seçme

Etmen öğrenme aşamasındayken, amacına ulaşamadığı her durum-eylem çifti için negatif değerlerle çevre tarafından cezalandırılır. Bu işlemdeki amaç; etmenin daha önce gitmediği durumlara gitmesi, aynı kareler üzerinde gezinmeyi bırakmasıdır. Bu işlem yardımıyla etmen, durum uzayını daha eşit miktarda öğrenebilmektedir. Öğrenme aşamasında yapılan diğer bir işlem ise etmenin q değerlerini güncellemekte kullandığı öğrenme katsayısını azaltmaktır. Bunun nedeni ise öğrenen etmenin daha fazla öğrenmeye ihtiyacı olmamasıdır. Etmen için öğrenme katsayısı başlangıçta bire yakınken sonlara doğru sıfır değerine daha yakın olmaktadır.

5.3. Sinir Ağıyla Hızlandırılmış Takviyeli Öğrenme Algoritması

Yapay sinir ağları kullanarak geliştirdiğim takviyeli öğrenme algoritmasında etmen, hakkında hiçbir şey bilmediği durum uzayına bırakılarak, amacı olan ödüle en kısa yoldan ulaşmaya çalışmaktadır. Etmenin hayatı süresince gerçekleştireceği hareketleri belirleyen algoritma adımları aşağıda gösterilmektedir:

1. Ağırlıklar rasgele değerlere atanır.

2. Daha önceden belirlenen eğitim örneklerine göre yapay sinir ağı eğitilir.

3. Etmenin içinde bulunduğu güncel durum belirlenir.

4. Algılayıcılar ve yapay sinir ağları yardımıyla etmenin bulunduğu durumda yapabileceği eylem kümesi belirlenir.

5. Aşağıdaki denklem kullanılarak etmenin gerçekleştireceği eylemin yönü seçilir.

$$action = \arg\max(Q_{nn}(s,a) + Q(s,a))$$

6. Etmen seçilen yöne doğru hareket ederek yeni konumuna gelir.

7. Etmen amacına ulaşmışsa çevre tarafından pozitif bir sayısal değerle ödüllendirilir, ulaşamamışsa negatif bir sayısal değerle cezalandırılır.

8. Gerçekleştirdiği durum-eylem çiftine ait q değerlerini aşağıdaki formülü kullanarak güncelleştirir.

$$Q(s,a) = (1-\beta)Q(s,a) + \beta(r(s,a) + \gamma \max_{a'} Q(s',a'))$$

9. Yapay sinir ağında gerçekleştirilen eylemle ilgili ağırlık değerleri aşağıdaki formüle göre güncelleştirilir.

$$w_{(i,j)} = w_{i,j} + \beta(r(s,a) + \gamma \max_{a'} Q(s',a'))$$

10. Etmen öğrenene kadar 3.-9.maddeler arası işlemler devam ettirilir.

5.4. Bilgisayar Simülasyonu ve Deneysel Sonuçlar

5.4.1. Bilgisayar simülasyonu

Yapay sinir ağlarıyla hızlandırılmış takviyeli öğrenme metodu bilgisayar simülasyonu, İntel Celeron 1.7, 512 Mbyte RAM'a sahip diz üstü bilgisayarda Delphi 7 yazılım geliştirme ortamı kullanılarak hazırlanmıştır.

Geliştirilen programda bir labirent ve arama problemi uygulanmaktadır. Engellerle karmaşık hale getirilmiş 15x15'lik bir ızgara içerisine bırakılan etmen, deneme yanılma metoduyla durum uzayını tanıyarak ödüle giden en kısa yolu bulmaya çalışmaktadır.

5.4.2. Deneysel Sonuçlar

Geliştirilen bilgisayar simülasyonunda değişik çevrelerde birçok deneyler yapılmıştır. İlk başta daha basit olan engelsiz çevrede deney yapılmıştır. Daha sonra engel sayıları ve böylece çevre karmaşıklığı da kademeli olarak arttırılarak deneylere devam edilmiştir. Yapılan deneylerde geliştirilen metot ile q öğrenme algoritması karşılaştırılmış ve önerilen metodun q öğrenmeye göre daha iyi sonuçlar verdiği gözlemlenmiştir.

Önerilen metoda uygun programlanmış bilgisayar simülasyonunda etmen gerçekleştirdiği her hareket sonrası çevreden sayısal değerde takviye sinyali (r) alır. Etmeni eğitmek için kullanılan ödül değerleri aşağıda gösterilmektedir:

- r = 100 etmenin amacına ulaştığı durumlar için
- r = -0.3 diğer durumlar için

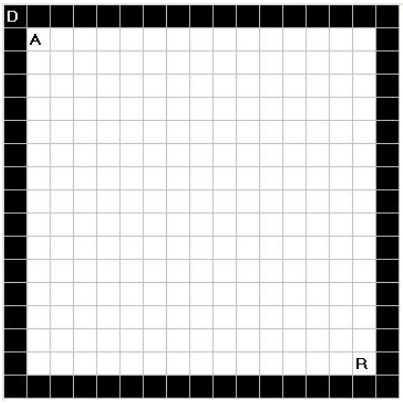
Etmeni eğitmek için kullanılan diğer parametreler de Tablo 5.1'de gösterilmektedir.

Parametre	Değer	Açıklama
β	0.8	Öğrenme katsayısı
γ	0.95	İndirgeme Oranı
ε	0.00001	Yapay sinir ağları hata payı

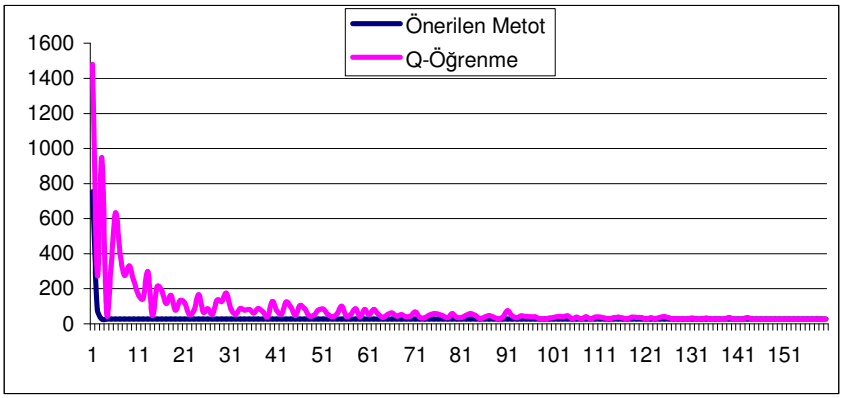
Tablo 5.2 Kullanılan Parametreler

Etmen ortam içinde amaçsız hareket ederek öğrenmeye başlar. Deneme yanılma yoluyla yaptığı hareketler sonrasında çıkışı bularak deneyimlerini arttırır. Çıkışı en kısa yoldan bulmasına yardımcı olacak q ve ağırlık değerlerini günceller. Yön seçme işleminde q değerleri kadar yapay sinir ağı çıkışları da dikkate alınır. Etmen amacına ulaştığı zaman program yeniden başlatılır ve öğrenme işlemine güncellenmiş verilerle devam edilir. Etmen optimum yolu bulduğu zaman program sonlandırılır. Böylece etmen amacına ulaşmış ve öğrenme işlemi sona ermiştir. Geliştirilen uygulamaya ait üç adet deneysel örnek ve sonuçları aşağıda gösterilmektedir:

5.4.2.1. Deneysel Örnek – 1

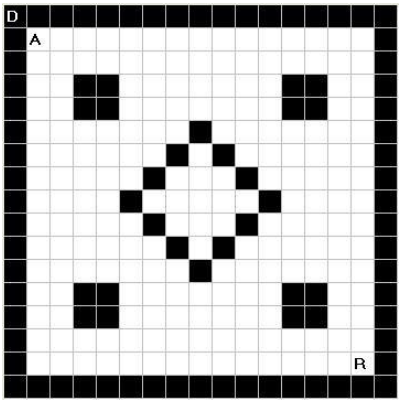


Şekil 5.7 Simüle edilmiş Çevre-1

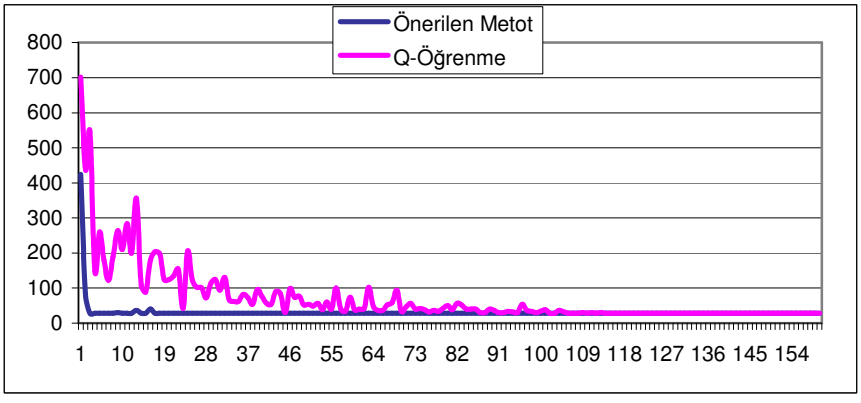


Şekil 5.8 Çevre-1 için önerilen metot ve Q öğrenme uygulama sonuçları

5.4.2.2. Deneysel Örnek – 2

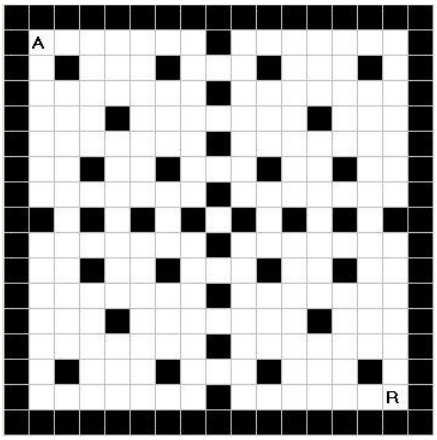


Şekil 5.9 Simüle edilmiş Çevre-2

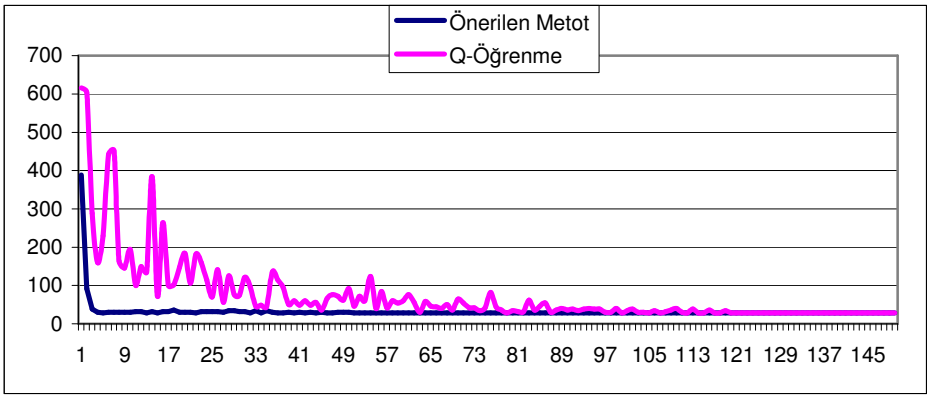


Şekil 5.10 Çevre-2 için önerilen metot ve Q öğrenme uygulama sonuçları

5.4.2.3. Deneysel Örnek – 3



Şekil 5.11 Simüle edilmiş Çevre-3



Şekil 5.12 Çevre-3 için önerilen metot ve Q öğrenme uygulama sonuçları

6. SONUÇ

Bu çalışmamda yapay sinir ağı kullanılarak geliştirilmiş takviyeli öğrenme metodunun labirent ve arama problemlerinde kullanılmasını anlattım. Sonuçlarında gösterdiği gibi bağımsız bir etmen hiçbir ön bilgiye sahip olmadığı çevrede bu yöntem kullanılarak daha hızlı bir şekilde öğrenebilmektedir. Etmen geliştirilen metodu kullanarak amacına en kısa yoldan ulaşma yeteneğine sahip olabilmektedir. Bu yöntem diğer bağımsız etmen uygulamaları içinde kabul edilebilir.

KAYNAKLAR

1. **Gerhard Weiss, 2000, "**Multiagent Systems - A Modern Approach to Distrubuted Artificial Intelligent"
2. **Hoshino Y., Kamei K., 2003,** "A Proposal of Reinforcement Learning System to Use Knowledge Effectively"
3. **Harmon E., Harmon S., 1998,** "Reinforcement Learning: A Tutorial"
4. **Barto, Sutton, & Watkins, 1990,** "Grefenstette, Ramsey, & Schultz"
5. **M. Wooldridge and N.R. Jennings, 1995,** " Intelligent agents: Theory and practice" , The Knowledge Engineering Review, 10(2):115-152
6. **Ben Coppin, 2000,** "Artıficial Intelligence Illuminated: Intelligent Agents"
7. **Mehmet KAYA,1998, "**Yüksek Lisans Tezi, "Akıllı Yazılım Etmenleri" Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ"
8. **S. Russell, P. Norvig, 1995, "**Artificial Intelligence: A Modern Approach", 46
9. **http://www.cs.berkeley.edu/~russel/aima.html**
10. **Kaelbling L.P., Littman M.L., & Moore A.W.,1996,** "Reinforcement learning: a survey. Journal of Artificial Intelligence Research", 4,237-285
11. **Tom M. Mitchell, 1997,** "Machine Learning: Reinforcement Learning" Carnegie Mellon University
12. **Sutton, R. S. and Barto, A. G., 1998,** Reinforcement Learning: An Introduction. Cambridge, MA: MIT Pres
13. **Abul,o., Polat,F. And Alhajj, R., 2000**, "Multiagent Reinforcement Learning using function approximation", IEEE transaction on System, Man and cybernetics-Part C,Vol. 30, No : 4
14. **Kaya M. 2003**, Doktora tezi," Çoklu etmen veri madenciliği" , Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ
15. **Watkins, C.J.C.H. and Dayan P., 1992,** "Technical Note: Q-Learning, Machine Learning Vol.8. pp.279-292"
16. **http://www.cs.ualberta.ca/~sutton/book/1/node7.html**

17. **Uchibe E., Asada M., Hosoda K., 1996,** "Behaviour Coordination for a Mobile Robot Using Moduler Reinforcement Learning" IEEE
18. **Elmas, Ç., 2003**, *"Yapay Sinir Ağları (Kuram,Mimari,Uygulama)"*, Seçkin Yayınları.
19. **Eşiyok, E., A. S.Yılmaz ve M. Bayrak., 1996,** "Yük Dağıtım Merkezlerinde Yapay Sinir Ağları ile Kısa Dönemli Yük Tahmin Analizi."
20. **Sağıroğlu Ş., Beşdok E., Erler M., 2003,** *"Mühendislikte Yapay Zeka Uygulamaları Yapay Sinir Ağları"*, Ufuk Yayınları.
21. **Yücetürk, A. C., 2000,** "Yapay Sinir Ağları Kullanılarak Örüntü Sınıflandırma ve Tanıma", Doktora Tezi, Ege Üniversitesi.
22. **Huang, B., Cao, G., Mın, G., 2005,** "Reinforcement Neural Network to the problem of Autonomous Mobile Robot Obstacle Avoidance"