

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

139657

TRENLERARASI ÇATIŞMALARIN ÇÖZÜMÜNDE
DİSPEÇER KARARLARININ YAPAY SİNİR AĞI
MODELİ

139657

İnşaat Müh. Selim DÜNDAR

FBE İnşaat Mühendisliği Anabilim Dalı Ulaştırma Programında
Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. İsmail ŞAHİN (YTÜ)

Prof. Dr. Aydın EREL

Prof. Dr. Haluk Bursalı

İSTANBUL, 2003

TC. YÜKSEK LİSANS TEZİ
DOKÜMANI

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	i
KISALTMA LİSTESİ.....	ii
ŞEKİL LİSTESİ.....	iii
ÇİZELGE LİSTESİ.....	iv
ÖNSÖZ.....	v
ÖZET.....	vi
ABSTRACT.....	vii
1. GİRİŞ: Amaç ve Kapsam.....	1
1.1 Amaç.....	1
1.2 Kapsam.....	3
1.3 Kaynak Araştırması.....	4
2. DEMİRYOLU TRAFİK YÖNETİMİNİN ESASLARI.....	7
2.1 Giriş.....	7
2.2 Demiryolu Trafik Yönetimi Problemi.....	8
2.3 Demiryolu Trafik Yönetiminin Hiyerarşik Analizi.....	10
2.3.1 Trafik Yönetim Merkezi.....	10
2.3.2 Trafik Yönetim Bölgesi.....	12
2.4 Demiryolu Trafik Yönetiminin İşlevsel Analizi.....	13
2.5 Yeniden Çizelgeleme ve Trafik Kontrolü.....	15
2.5.1 Önceden Hazırlanmış Çizelge ve Gerçekleşen Çizelge.....	15
2.5.2 Yeniden Çizelgeleme ve Trafik Kontrolü için Gerekli Bilgiler.....	17
2.5.3 Yeniden Çizelgeleme ve Trafik Kontrolüne Yönelik Bazı Kurallar.....	19
3. YAPAY SİNİR AĞLARI.....	21
3.1 Giriş.....	21
3.2 İnsan Beyninin Temel Çalışma Prensipleri.....	21
3.3 Temel Yapay Sinir Ağı Modelleri ve Öğrenme Kuralları.....	22
3.3.1 Nöronlar.....	24
3.3.2 Bağlantılar.....	26
3.3.3 Öğrenme Kuralları.....	27
3.3.3.1 Hebb Öğrenme Kuralı.....	29
3.4 Tek Katmanlı İleri Beslemeli Ağlar.....	30
3.4.1 Perseptron Öğrenme Kuralı.....	31
3.4.2 Adaline.....	34
3.5 Çok Katmanlı İleri Beslemeli Ağlar.....	37
3.5.1 Geri Yayma Algoritması.....	37
3.5.2 Geri Yayma Algoritmasının Öğrenme Faktörleri.....	46
3.5.2.1 Başlangıç Ağırlıkları.....	46
3.5.2.2 Öğrenme Sabiti.....	46
3.5.2.3 Değer Fonksiyonu.....	47
3.5.2.4 Momentum.....	48
3.5.2.5 Güncelleme Kuralları.....	49
3.5.2.6 Eğitim Verisi ve Genelleştirme.....	50
3.5.2.7 Saklı Nöron Sayısı.....	50

4.	TRENERARASI ÇATIŞMALARIN ÇÖZÜMÜ İÇİN YAPAY SİNİR AĞI MODELİ..	52
4.1	Giriş.....	52
4.2	Dispeçerin Karar Değişkenleri.....	53
4.2.1	Temel Öncelik Sayısı.....	54
4.2.2	Kritik Oran.....	54
4.2.3	Miyopik Çözümdeki Gecikme.....	54
4.2.4	Potansiyel Çatışma Sayısı.....	55
4.3	Yapay Sinir Ağı Modelleri.....	55
4.3.1	Normalizasyon.....	56
4.3.2	Tek Katmanlı İleri Beslemeli Model.....	56
4.3.3	Çok Katmanlı İleri Beslemeli Model.....	57
4.4	Sayısal Uygulama Sonuçlarına İlişkin Bazı Saptamalar.....	60
5.	SONUÇLAR ve ÖNERİLER.....	64
	KAYNAKLAR.....	66
	ÖZGEÇMİŞ.....	67



SİMGE LİSTESİ

$a(f)$	Nöronun aktivasyon fonksiyonu.
$d^{(k)}$	k paterni için istenen çıkış değeri.
$E(w)$	Değer fonksiyonu.
E^∞	Hata vektöründeki en büyük eleman.
F	Giriş uzayının boyutu.
i^*	Çıkış vektöründeki en büyük elemanın değeri.
k_i	Nöron i 'ye bağlanan giriş sayısı.
M	Lineer sağımsız bölge sayısı.
N_m	Gerekli saklı nöron sayısı.
$net_i^{(k)}$	Nöron i 'ye net giriş.
P	Eğitme paterni sayısındaki alt sınır.
r	Öğrenme sinyali.
w_{ij}	Nöron j 'yi, nöron i 'ye bağlayan ağırlığın değeri.
$x^{(k)}$	k giriş paterni.
$x_i(t)$	t anındaki i giriş vektörünün değeri.
$y^{(k)}$	k paterni için elde edilen çıkış değeri.
$y_i(t)$	Nöron i 'nin t anındaki çıkış değeri.
z_q	Saklı katmanın çıkış değeri.
α	Momentum değeri.
δ_{hq}	Saklı katmandaki nöron q 'nun hata sinyali.
δ_{oi}	Çıkış katmanındaki nöronların hata sinyali.
Δw_{ij}	Nöron j 'yi, nöron i 'ye bağlayan ağırlığın değerindeki değişim miktarı.
η	Öğrenme sabiti.
θ_i	Nöron i 'nin eşik değeri (bias).
ρ	Kürenin yarıçapı.

KISALTMA LİSTESİ

ADALINE	Adaptive Linear Element (Eğitilebilir Doğrusal Eleman)
CTC	Centralized Traffic Control (Merkezi Trafik Kontrolü)
M.Ö.	Milattan Önce
M-P	McCulloch&Pitts nöronu
O.S.	On Switch (Makastan geçiş)
TCDD	Türkiye Cumhuriyeti Devlet Demiryolları
TMİ	Trafiğin Merkezden İdaresi



ŞEKİL LİSTESİ

Şekil 2.1	Merkezi trafik kontrolünde demiryolu trafik yönetiminin elemanları arasındaki (hiyerarşik) ilişki.....	11
Şekil 2.2	A-G Demiryolu kesimine ait önceden hazırlanmış çizelge.....	16
Şekil 2.3	Planlanmamış durum ve ortaya çıkan trenlerarası çatışmalar.....	16
Şekil 2.4	Trafik kontrolünde 'geribesleme'.....	17
Şekil 3.1	Tipik bir sinir hücresi örneği.....	22
Şekil 3.2	McCulloch&Pitts nöronu.....	23
Şekil 3.3	Bazı aktivasyon fonksiyonları. (a) Basamak fonksiyonu. (b) Eşik fonksiyonu. (c) Rampa fonksiyonu. (d) Unipolar sigmoid fonksiyonu. (e) Bipolar sigmoid fonksiyonu.....	25
Şekil 3.4	Basit bağlantı geometrileri. (a) Tek katmanlı ileri beslemeli ağ. (b) Çok katmanlı ileri beslemeli ağ. (c) Kendi kendini besleyen nöron. (d) Tek katmanlı geri beslemeli ağ. (e) Çok katmanlı geri beslemeli ağ.....	27
Şekil 3.5	Üç öğrenme türü. (a) Eğitici öğrenme. (b) Takviyeli öğrenme. (c) Eğitici öğrenme.....	28
Şekil 3.6	Genel öğrenme yapısı (eğitici öğrenmede d _i bulunmaz).....	29
Şekil 3.7	XOR problemi.....	31
Şekil 3.8	Basit bir perseptron.....	32
Şekil 3.9	Perseptron öğrenme kuralında ağırlık güncellemeleri.....	33
Şekil 3.10	Üç katmanlı, geri yaymalı ağ örneği.....	38
Şekil 3.11	Farklı sayıdaki saklı birimle, geri yayma algoritmasındaki yakınsama.....	42
Şekil 3.12	XOR probleminin çözümünde kullanılan ağ yapısı.....	43
Şekil 3.13	Basit bir düzlemde eğitim düşümleri. Solda momentum terimi yoktur, sağdakilerde vardır.....	48
Şekil 4.1	Tren i'nin miyopik çözümdeki gecikmesi.....	55
Şekil 4.2	Tek katmanlı ileri beslemeli ağ modelinde eğitim düşümü.....	57
Şekil 4.3	Saklı katmandaki tek nöronla eğitim düşümü.....	57
Şekil 4.4	Saklı katmandaki 16 nöronla eğitim düşümü.....	58
Şekil 4.5	10 Girişli modelde saklı katmandaki 20 nöronla eğitim düşümü.....	59
Şekil 4.6	8 Giriş parametresi, 16 saklı nöron, 1 çıkış nöronuna sahip model.....	60

ÇİZELGE LİSTESİ

Tablo 4.1	XOR problemi için çıkış değerleri.....	32
Tablo 5.1	Temel öncelik sayıları.....	54
Tablo 5.2	Modele dahil edilen parametrelerin ağ performansı üzerinde etkileri.....	60
Tablo 5.3	Trenlerarası çatışma örnek ve çözümleri.....	61



ÖNSÖZ

Öncelikle, bilgi ve deneyimlerinden sürekli yararlandığım, görüşleri bana sürekli ışık tutan, bu çalışmayı gerçekleştirebilmemde hiçbir katkıyı esirgemeyen, Danışmanım Yrd. Doç. Dr. İsmail Şahin'e;

Lisansüstü çalışma yapmam konusunda beni teşvik eden ve yardımlarını esirgemeyen Prof. Dr. Sönmez Yıldırım'a; lisansüstü çalışmam boyunca bana yardımcı olan, bilgi dağarcığıma yeni bilgiler girmesini sağlayan, Prof. Dr. Aydın Erel, Prof Dr. Zerrin Bayrakdar, Prof Dr. Mustafa Ilıcalı ve Öğr. Gör. Dr. Halit Özen'e;

Verdiği ders ile yapay sinir ağları konusunu öğrenebilmemi sağlayan, Doç. Dr. Tülay Yıldırım'a;

Yıldız Teknik Üniversitesi, İnşaat Mühendisliği Bölümü Ulaştırma Anabilim Dalı'nda görevli bulunan, lisansüstü çalışmam boyunca yardımlarından ve arkadaşlıklarından sıkça yararlandığım, Öğr. Gör. Dr. Nilgün Camkesen, Öğr. Gör. Dr. Mustafa Sinan Yardım, Araş. Gör. Mustafa Gürsoy, Araş. Gör. Haluk Yüksel ve Araş. Gör. Güzin Akyıldız'a;

Bilgisayar çalışmalarım sırasında yardımlarını gördüğüm, arkadaşım İnş. Yük. Müh. Murat Emre Batmaz'a;

Çalışma içerisinde bulunan şekillerin son halini almasında katkılarından dolayı, arkadaşım Desinatör Gürcan Eralp'e;

Çalışmada kullandığım MATLAB adlı programı öğrenmemi sağlayan ve bu program için yazdığı komutlarla işlerimi kolaylaştıran, arkadaşım Elektronik Yük. Müh. Ebru Köksaldı'ya;

Uygulamalarda kullandığım bilgilere erişmemde sağladıkları kolaylıktan dolayı TCDD Birinci Bölge Başmüdürlüğü'ne;

Ve en önemlisi, beni her konuda teşvik eden, desteklerini esirgemeyen, geldiğim noktada inkar edilemeyecek katkıları bulunan çok sevgili anneme, babama ve anneanneme;

içten teşekkürlerimi sunarım.

ÖZET

Bir ulařtırma alt sistemi olan demiryolu, günümüzde diđer ulařtırma alt sistemleriyle yoğun bir rekabet ortamındadır. Rekabet edebilmenin en önemli kořullarından biri sunulan hizmetin kalitesidir. Bir ulařtırma alt sisteminin kalitesine iliřkin önemli iki ölçüt dakiklik ve güvenilirliktir. Bu ölçütlerin sürekli istenilen düzeyde tutulabilmesi, etkin trafik kontrolü ile sağlanabilir.

Bu çalışmada, trafiğin önceden hazırlanmış çizelgeye uygun yürütüldüğü plan duyarlı işletmecilik kořullarındaki trafik kontrol problemi ele alınacaktır. Önceden hazırlanmış çizelge, trenlerarası çatıřmalar bulunmayan bir plandır. Ancak planlanmamış olayların gerçekteleřmesiyle trenlerarası çatıřmalar meydana gelebilir. Trafik kontrolü, trenlerarası çatıřmaları, gecikmeleri minimize ederek çözmeyi hedefleyip, yeni bir çizelge hazırlamak için uygulanır. Problemin büyük boyutlu ve karmařık olması nedeniyle, optimum çözüme makul bir müddet içerisinde ulařılması imkansızdır. Dolayısıyla, tren dispeçerleri çatıřma çözümlerinde bazı kurallar kullanarak problemi basitleřtirmektedirler. Bu çalışmada, tren dispeçerlerinin trenlerarası çatıřmalarda verdikleri kararlar incelenmiş ve yapay sinir ağı tekniğıyle bu kararları modellenmiştir. Gerçekteleřtirilen sayısal testlerde modelin büyük bir başarı oranıyla çalıştığı, yani dispeçerin kararlarını yüzde 92 oranında taklit edebildiğı saptanmıştır.

İkinci bölümde, demiryolu trafik kontrolünün esasları ortaya konmuş ve yeniden çizelgeleme üzerinde durulmuřtur.

İzleyen bölümde, yapay sinir ağı tekniğı ve ağların genel çalışma prensipleri incelenmiştir. Yapay sinir ağı tekniğinin elemanları tanıtıldıktan sonra, kullanılan farklı eğitime yöntemlerine değinilmiştir.

Dördüncü bölümde, trenlerarası çatıřmaların çözümünde dispeçerin verdiğı kararları taklit eden bir yapay sinir ağı modeli oluşturulmuřtur. Modelde çok katmanlı ileri beslemeli ağ yapısı ve öğrenme için geri yayma algoritması kullanılmıştır. Model, daha önce gerçekteleřmiş çatıřmalarda, dispeçerin verdiğı kararları inceleyip, gelecekteki çatıřmalarda da benzer kararları verebilecek şekilde eğitilmiştir. Modelde, dispeçerin kararlarında, *temel öncelik sayısı, kritik oran, miyopik çözümdeki gecikme ve potansiyel çatıřma sayısını* göz önünde bulundurduğı kabul edilmiştir.

Çalışmanın son bölümünde, geliřtirilen modelin sonuçları ve öneriler sunulmuřtur.

Anahtar kelimeler: Tren çizelgeleme, demiryolu trafik kontrolü, yapay sinir ağları, geri yayma algoritması, çok nitelikli karar verme.

ABSTRACT

Railways, as a transportation sub-system, is in a competition with other subsystems. One of the most important attributes of competition is the quality of the service provided. Two important criteria about the quality of a transportation sub-system are timeliness and reliability. Keeping these criteria at a certain desired level can be achieved by performing effective traffic control.

In this study, we deal with traffic control problem for schedule-responsive operations, which is carried out based on a pre-determined schedule. Train conflicts may occur when trains deviate from this schedule. Traffic control procedures are performed for rescheduling train movements to minimize delays of trains running in the system. In this study, decisions made by train dispatchers have been examined and modelled using artificial neural networks technique. We have performed several numerical tests to determine the success of the model developed. The test results showed that proposed neural network model can reproduce the decisions of train dispatchers with a 92 percent success rate.

In chapter two, the railway traffic management process is outlined.

We then introduce the artificial neural networks and explain the way they work in chapter three. Having defined the elements of artificial neural networks, we detailed the various network types.

A neural network model, which is developed to imitate the dispatcher's decisions in solving train conflicts, is presented in the subsequent chapter. The model was provided with numerous decisions given by train dispatchers in the past and trained to reproduce similar decisions when a new conflicting situation is introduced. Several numerical test were performed in order to identify the success of the model and the effects of various input parameters on the outcome. The test results are given at the end of the chapter.

In the last chapter of the study, conclusions and some future prospects are presented.

Keywords: Train scheduling, train dispatching, artificial neural networks, back propagation algorithm, multiple criteria decision making.

1. GİRİŞ: Amaç ve Kapsam

1.1 Amaç

Ulaştırma sisteminin alt sistemlerinden biri olan demiryolu, yüksek taşıma kapasitesi ve düşük işletme maliyeti ile, diğer ulaştırma alt sistemlerine kıyasla büyük bir avantaja sahiptir. Yine de diğer ulaştırma alt sistemleri ile rekabet edebilmesi için, etkin ve etkili bir işletmecilik yapılması gerekmektedir. Sunulan hizmetin kalitesi, demiryollarının diğer alt sistemlerle yarışabilmesi için, çok önemli bir kıstastır. Etkin ve etkili (verimli) işletmecilik, hizmet kalitesini çok önemli ölçüde etkilemektedir. Etkin işletmecilik, önceden belirlenmiş hedeflere ulaşmakla gerçekleşebilir. Bu hedeflerin en önemlilerinden biri, trenlerin istasyonlara önceden belirlenmiş varış zamanlarında ulaşmalarıdır. Etkili işletmecilik ise, mevcut demiryolu kaynaklarını (örneğin, personel, ekipman, zaman, vb.) en verimli şekilde kullanılması demektir. Sunulan hizmetin kalitesi, tren seferlerinin dakikliği ve güvenilirliği ile açıklanabilir. Burada dakiklik, tren hareketlerinin önceden hazırlanmış çizelgeye uygun olarak düzenlendiği plan duyarlı işletmelerde, planlanmış zamanlara uygunluk olarak belirtilebilir. Güvenilirlik ise, bu zamanlardan sapma miktarı olmaktadır.

Günümüzde yük ve yolcu taşımacılığı büyük bir rekabet ortamında gerçekleştirilmektedir. Müşteriler açısından, varış zamanlarının bilinmesi ve güvenilirliği büyük önem taşımaktadır. Bu iki etken, tür seçimini büyük ölçüde etkilemektedir. Bu yüzden, demiryolu işletmeleri, başlangıç ve son noktaları arasındaki taşımacılığı zamanında, ekonomik ve güvenilir olarak gerçekleştirmek zorundadırlar. Bu sebeplerden dolayı, plan duyarlı işletmecilik, demiryollarında çok büyük önem taşır.

Demiryolu ulaşımında, seyir sürelerindeki değişim, varış zamanlarını doğrudan etkilemektedir. Bu yüzden, önceden planlanmış çizelgeye uygun hareket tehlikeye girebilmektedir. Böyle bir durumda, bağlantısı olan ya da aktarma yapacak olan seferler aksayabilmekte, tren personelinin yasal çalışma süresi aşılabilmekte ve geciken trenin lokomotif ve vagonları verimli kullanılamamaktadır. Seyir sürelerindeki sapmanın kabul edilebilir sınırlar içinde kalması, ancak etkin bir işletmecilik anlayışı ile gerçekleştirilebilir. Bu da demiryolu işletmeciliğinin önemli bir parçası olan tren çizelgeleme ve trafik kontrol işlemlerinin araştırılması için önemli bir hareket nedeni oluşturmaktadır.

Etkin bir işletmecilik için, demiryolu trafiğinin amaca uygun kontrol edilmesi gereklidir. Trafik kontrolü problemi, trenlerin önceden hazırlanmış çizelgeden sapma durumunda karşımıza çıkmakta, bunun sonucunda da trenlerarası çatışmalar meydana gelmektedir. Bu çalışmada, trenlerarası çatışma, iki trenin demiryolu hattı üzerinde, istasyon ve saydingler dışında, aynı yerde aynı zamanda bulunması durumunu ifade etmektedir. Tek hatlı bir demiryolunda, zıt yönde hareket eden iki trenin, iki komşu istasyon arasında karşılaşması bir çatışma durumudur. Böylece, trafik kontrol problemi; çatışan tren hareketlerinin çözüleceği istasyon/saydinglerin belirlenmesi ve herhangi bir anda hangi trenin hangi hatta bulunması gerektiğinin saptanmasıdır. Burada, kontrole ilişkin amaçlar, tren gecikmelerinin en küçüklenmesi, tren hareketlerinin güvenliği ve etkili hat kapasitesi kullanımı olmaktadır.

Demiryolu trafik kontrolü günümüzde deneyimli trafik kontrolörleri (dispeçerler) vasıtasıyla yürütülmektedir. Dispeçerler, yıllar boyunca kazandıkları bilgi ve deneyimleri vasıtasıyla, karar verme yeteneklerini kullanarak, trenlerarası çatışmaları çözmektedirler. Bunu yaparken de doğal olarak problemi basitleştirme yolunu seçmektedirler. Örneğin, trenlere öncelik sırasına göre hat kesimlerini kullanma izni vermektedirler. Öncelik kuralları, belirli ölçüde kolaylık sağlarken, optimum çözümden uzaklaşma gibi bir dezavantajı da vardır. Sunulan hizmetin düzeyi ile optimum çözümün ilişkisi düşünüldüğünde, kullanılan öncelik kurallarının akılcı olması ve optimuma yakın çözümler üretmesi gerekmektedir. Bu da ancak trenlere ait önceliklerin sürekli güncellenmesi ile mümkündür. Yani trenlerin erken ya da geç seyirleri durumunda, bunların öncelikleri yenilenmelidir.

Bu, çalışmanın amacı, yukarıda belirtilen koşullar altında ortaya çıkan bir trenlerarası çatışma durumunda, dispeçerin kararlarını taklit edebilen bir sistem geliştirmektir. Bu, örnekler yoluyla öğrenme özelliğine sahip bir sistemdir. Dispeçerin geçmişte verdiği kararlar örnek kümesini oluşturur. Bu örneklerde, dispeçerin ne durumda, hangi kararları verdiği bilinmektedir. Diğer bir deyişle, dispeçerin kararlarında dikkate aldığı kabul edilen tren özelliklerinin (kriterlerinin) sayısal değerleri ile iki trenin çatışması durumunda (bu trenlerin özellikleriyle birlikte), dispeçerin hangi trenin geçmesine izin verdiği ve hangisini beklettiği bilinmektedir. Bu sistem, öncelikle dispeçer kararlarını inceleyerek, çatışma durumlarında hangi kriterlere daha çok önem verdiği saptayıp, daha sonra gerçekşelecek çatışmalarda da dispeçerin vereceği muhtemel kararı öngörececek bir model içermektedir. Burada geliştirilen

sistemde, çok katmanlı yapay sinir ağı modeli ve bir öğrenme algoritması (hatayı geri yayma algoritması) kullanılmıştır. Elde edilen sonuçlar, tatmin edicidir.

1.2 Kapsam

Çalışmanın ikinci bölümünde, demiryolu trafik yönetiminin esasları açıklanmıştır. Burada, trafik yönetimi probleminin tanımı yapılmakta; yönetimin eleman ve alt elemanları tanımlanmakta ve aralarındaki hiyerarşik ilişki ortaya konmaktadır. Daha sonra, trafik yönetiminin en önemli işlevlerinden biri olan 'yeniden çizelgeleme' ve 'trafik kontrolü' ele alınarak, kurulan modelin daha iyi anlaşılabilmesi için bir hazırlık yapılmaktadır.

Üçüncü bölümde, yapay sinir ağı sistemleri açıklanmaktadır. Burada, yapay sinir ağlarının tanımı yapılmakta ve esinlenme kaynağı olan insan beyninin temel çalışma prensipleri açıklanmaktadır. Daha sonra, temel sinir ağı modelleri ve öğrenme kurallarına değinilerek, basit bir yapay sinir ağının nasıl çalıştığı anlatılmaktadır. Tek katmanlı ileri beslemeli ağlara değinilerek, lineer çözüme sahip problemlerin modellenmesinin açıklanmasının ardında çok katmanlı ağ yapılarına değinilmiştir. Çok katmanlı ileri beslemeli ağlar, lineer çözümü mümkün olmayan problemlerin modellenmesi için faydalı bir alternatiftir. Burada, trenlerarası çatışmalarda dispeçer kararlarını modellemek için kullanılacak olan, geri yayma algoritması açıklanmış ve bu algoritmanın öğrenme faktörlerine değinilmiştir.

Dördüncü bölümde, dispeçerin karar davranışının yapay sinir ağı modelinin oluşturulması amacıyla geri yaymalı bir ağ yapısı oluşturulmuş ve modelin çözümü yapılmıştır. Modelde, trenlere ait (trafik kontrolünde önemli olduğu kabul edilen) dört adet nitelikten yararlanılmıştır. Bu niteliklerin, her trenin çatışma anındaki durumunu en iyi tanımlayan nitelikler olmasına dikkat edilmiştir. Bunun için, tren dispeçerleriyle yüzyüze görüşülmüş ve yaptıkları işler gözlenmiştir. Bu konuda, yapay sinir ağı yöntemini ilk defa kullanan böyle bir çalışmanın sonuçları oldukça olumludur. Gözlem ve deneyimlere dayanılarak oluşturulan yapay sinir ağı modeli, dispeçerin karar davranışlarını yüzde 92 başarıyla taklit edebilmektedir.

Beşinci ve son bölümde, çalışmanın genel bir değerlendirmesi yapılmakta ve problemin kapsamının genişletilmesine ilişkin bazı öneriler sunulmaktadır.

1.3 Kaynak Araştırması

Demiryolu trafik kontrolü probleminin, çözümüne ilişkin çalışma sayısı oldukça azdır. Bunun en önemli nedenlerinden biri, problemin optimum çözümünün klasik matematik teknikleri kullanarak bulunmasının neredeyse imkansız olmasıdır. Sezgisel algoritmalar ile bulunan çözümler ise yaklaşık olmaktan öteye gidememiştir. Trafik kontrol probleminde önemli bir unsur teşkil eden dispeçer kararlarının ise nasıl (neye dayanılarak) alındığı, araştırmacılar tarafından çoğunlukla gözardı edilmiştir. Aşağıda, dispeçer kararlarının modellenmesi ve tren çizelgeleri ile ilgili yapılmış çalışmalar incelenmiş, böylece problemin çözümüne ilişkin yaklaşımlar belirlenmiştir.

Shen ve diğ. (1993), bir kurye servisindeki önemli mektupların gerçek zamanlı dağıtımında dispeçer kararlarını incelemişlerdir. Burada, dispeçerin yaptığı iş, her yeni gelen dağıtım işini, mevcut dağıtıcılardan birine atamaktır. Müşterilerin tercihleri, paketin teslim alındığı ve edildiği zamanlardır. Bu problemin çözümünde dispeçer, işletme maliyeti ve müşteri memnuniyeti gibi çatışan iki amaç arasında bir optimizasyon yapmak zorundadır. Shen ve arkadaşları, usta bir dispeçerin kararlarını modelleyen bir adaptif (öğrenebilen) sinir ağı modeli kurmuşlardır.

Potvin ve diğ. (1995), kurye servisindeki mektupların dağıtılmasını incelemiş ve doğrusal programlama modeli ile yapay sinir ağları tekniğini kullanarak, dispeçer kararlarını modellemeye çalışmışlardır. Bu çalışmada, doğrusal programlama modeli %38, yapay sinir ağı ise %78 başarıyla dispeçerin kararlarını taklit edebilmiştir.

Vukadinovic ve diğ. (1996), dispeçer kararlarını Belgrad banliyö metro sisteminde incelemişlerdir. Burada dispeçerin amacı, yolcuların kayıplarını minimize etmektir. Bunlar bekletilen yolcu sayısının minimizasyonu ve toplam yolcu gecikmesidir. Bunun yanında dispeçer istasyonların önem sıralarını da gözönünde bulundurmak zorundadır, çünkü bazı istasyonlar diğerlerine göre daha üstündür. Ayrıca, tren sayısındaki yetersizlik de yolcuların metrodan başka türleri seçmesinde önemli rol oynamaktadır. Bu çalışmada, iki katmanlı ileri beslemeli adaptif bir yapay sinir ağı yapısı kullanılmış ve bu sistem dispeçer kararlarını taklit etmede %67 oranında başarılı olmuştur.

Şahin (1996), TCDD’de, trenlerarası çatışmaların en yoğun olarak gözlendiği Arifiye-Eskişehir arasındaki tek hatlı kesimde oluşan trenlerarası çatışmalarda dispeçer kararlarını incelemiştir. Bu çalışmada doğrusal programlama tekniği kullanılmış; kurulan model, %82 başarıyla, 90 çatışmanın 74’ünde dispeçerin karar davranışını aynen taklit etmiştir.

Vukadinovic ve diğ. (1997), iç denizlerdeki su taşımacılığında, yükleme, taşıma ve boşaltma işlemlerini modellemiştir. Sistemde dispeçerin verdiği karar, farklı kapasitelerdeki mavnalara, değişik boyutlardaki yükleri atamaktır. Bu çalışmada iki katmanlı ileri beslemeli adaptif bir ağ yapısı kullanılmış ve sistem, dispeçer kararlarını %95 başarıyla taklit etmiş ve dispeçerden daha iyi çözümler sunabilmiştir.

Higgins ve diğ. (1997), demiryollarının tek hatlı kesimlerinde gerçekleşen trenlerarası çatışmaların çözümü için, bazı sezgisel algoritmalar geliştirmişlerdir. Yerel arama sezgisel algoritması, genetik algoritmalar ve iki adet melez (birden fazla algoritmanın birarada kullanılması) algoritma geliştirerek, örnekelerin %90’ında, optimum çözümü %5’lik bir hata payı ile yakalayabilmişlerdir.

Tren rotalama ve çizelgeleme hakkında bilinen en son ve kapsamlı araştırma Cordeau ve diğ. (1998) tarafından yapılmıştır. Bu çalışmada, iyi bilinen analitik hat modelleri ve trafik kontrolü modellerini incelemiştir. Makalede de anlatıldığı gibi, trafik kontrol modelleri genellikle, tren gecikmelerini minimize etmek amacıyla trenlerin nerede karşılaşır, öne geçeceğini belirlemekle ilgilidir. Makalede, “... gerçek veri kümeleriyle test edilmiş olan en gelişmiş modellerin bile pek azı demiryolu trafik kontrolünde kullanılabilir şekilde geliştirilmiştir. Bu yüzden, teorik ve pratik arasındaki farkı kapatmak için çalışmalar yapılmalıdır.” ifadesi yer almaktadır.

Malavasi ve Ricci (2001), demiryolu trafik sirkülasyonu problemini yapay sinir ağları kullanarak modellemiştir; sisteme ilişkin kritik davranış parametrelerini kullanan -kendine- öğrenen bir sistem geliştirmişlerdir. Kullanılan kritik davranış parametreleri arasında; toplam demiryolu hattı uzunluğu, çift hatlı kesim uzunluğu, ray devreleri (otomatik

blok sinyal sistemi devreleri) sayısı, makas sayısı, hemzemin geit sayısı, referans alınan zaman içerisinde alışan yolcu treni ve yük treni adedi vardır.

Trenlerarası çatışmaların özümü için, dispeer kararlarının modellenmesinde yapay sinir ağı tekniğı, bildiğimiz kadarıyla daha önce hiç kullanılmamıştır. Bu nedenle bu alışma probleme yeni bir bakış açısı sunmaktadır.



2. DEMİRYOLU TRAFİK YÖNETİMİNİN ESASLARI

2.1 Giriş

Demiryolu araçları, izledikleri taşın yolu üzerinde tek boyutlu bir hareket olanağına sahiptir. Bu yüzden taşıt hareketlerinde güvenliğin sağlanması önemli bir işlemdir. Demiryolu hatları, taşıtlar tarafından ortaklaşa kullanıldığı için, etkili taşıt hareketleri vasıtasıyla, etkin bir işletmecilik yapılması gerekmektedir. Bu etkin ve etkili hareketler, demiryolu trafik yönetiminin en önemli hedeflerinden biridir. Demiryolu trafik yönetimi, uygun araçlar (veya kaynaklar, örneğin personel, taşıtlar, zaman, vb.) kullanılarak, istenilen hedeflere (örneğin güvenli ve etkin taşıt hareketlerini gerçekleştirmek) erişmek için yapılan işlemler bütünüdür. Sözü edilen kaynaklar, girdi olarak düşünüldüğünde, varılan hedefler çıktı olmaktadır. Demiryolu trafik yönetiminin başarı derecesi, çıktıların girdilere oranıdır ve bu da işletmenin verimliliğinin bir göstergesidir. Demiryolu kaynaklarının ne derece iyi kullanıldığı ölçütü verimliliktir. Ancak, bu çalışma kapsamında üzerinde durulan, istenilen hedeflere ne derece erişildiğinin bir göstergesi olan etkinlik olacaktır.

Çizelgeler, demiryolu trafik yönetiminde büyük öneme sahiptir. Tren hareketlerinin şematik olarak gösterildiği çizelge bir yol-zaman grafiğidir. Yatay ekseninde zaman ve dikey ekseninde ise yol (istasyon/saydın'lar) işaretlenir. Tren hareketleri, çizelge üzerinde eğik olarak çizilmiş doğru parçaları ile gösterilir. Yatay çizgiler de beklemleri ya da duruşları gösterir. Belirli uzunluktaki dönemler için hazırlanan çizelgelerin oluşturulması için bir ön çalışma yapılmalıdır. Sistemin kullanıcıların sosyo-ekonomik ve kültürel yapısı ve işletmenin kaynakları gözönünde bulundurularak hazırlanan çizelgeler; sunulan hizmetin belirli bir kalitede olmasıyla birlikte, işletme maliyetlerinin de minimizasyonu amaçlanarak oluşturulur. Bu bilgiler sayesinde, işletilecek tren türleri belirlenir ve bu trenlere kaynak (örneğin, personel, lokomotif, vagon vd.) ataması yapılır. Daha sonra, her tren için hareket planı oluşturulur. Hareket planı, her trenin ilk kalkış istasyonundan hareket zamanı; durması istenen istasyonlara varış ve/veya bu istasyonlardan kalkış zamanlarıyla bu istasyonlardaki duruş süreleri; ardışık istasyonlar arasındaki tabii ve en küçük seyir süreleri; ve son varış istasyonuna varış zamanı bilgilerini içerir. Belli bir demiryolu hattı üzerinde hareket edecek trenleri hareket planlarının bir araya getirilmesiyle çizelge (orer) oluşturulur. Bu çizelgelerde trenlerarası çatışmalar bulunmaz ve karşılaşma/önegeçme yer ve zamanları da belirlenmiştir.

Taktik (orta vadeli) anlamı olan bu çizelgeler, üst düzey yöneticilerinin katkısıyla, deneyimli işletme ve planlama personelinin ortak çalışması sonucu kesin şeklini alır.

Ulaşım hizmetlerinin düzenli olarak sunulduğu demiryolu işletmelerinde, bu hizmet önceden hazırlanmış çizelgeye uygun olarak yürütülür. Bu yöntem, *plan duyarlı işletmecilik* olarak adlandırılır. Sunulan hizmetin etkinlik düzeyi, bu çizelgeye uygunluğun bir göstergesidir ve tren hareketlerindeki gecikmelerle ölçülmektedir. Önceden hazırlanmış çizelgeye uygun olarak yürütülen işletmecilik, planlanmamış bir olay meydana gelene kadar devam eder. Planlanmamış olayların ortaya çıkmasıyla, plan duyarlı işletmecilik kuralları yerini *talep duyarlı işletmecilik* kurallarına bırakır.

Bu bölümde, demiryolu trafik yönetimi probleminin tanımı verilmekte, problemin hiyerarşik ve işlevsel analizi yapılmakta ve trafik yönetiminin ana işlevleri olan yeniden çizelgeleme ve trafik kontrolü işlemleri üzerinde durulmaktadır.

2.2 Demiryolu Trafik Yönetimi Problemi

Bir işletmeye ait demiryolu şebekesi, belli fiziki sınırlara sahip olan alt bölgelere ayrılır. Her alt bölge, trafik yönetimi açısından bir diğerinden bağımsızdır. Ancak, özellikle anahat trenlerinin, ilk kalkış ve son varış istasyonları arasında bulunan ardışık yönetim bölgelerinden geçmesi sebebiyle, bölgeler arasında sürekli bilgi iletişimi vardır. Her bölgede bir trafik yönetim merkezi bulunur. Bölge içindeki tren hareketleri bu merkez tarafından gözlenmektedir. Bölgeye giren ve bölgeden çıkan trenlerle, ara istasyon/sayding'lere varan ve bu noktalardan kalkan trenlerin numaraları, varış-kalkış yer ve zamanları düzenli olarak kaydedilir. Böylece, herhangi bir anda bir trenin nerede bulunduğu kolaylıkla bilinebilir. Bu bilgiler, trenlerin o ana kadar gerçekleşen hareket bilgileriyle birleştirilir ve trenlerin bundan sonraki hareketlerine ilişkin tahmin yapılmasında kullanılır. Trenlere ait bu bilgiler, trafik kontrol sistemi ya da haberleşme donanımı aracılığıyla yönetim merkezine iletilir. Tren hareketlerinin kontrol işleminin (örneğin bir trenin istasyon/sayding yan hattına alınması için yol tanzimi -tren rotalaması) uzaktan kumanda edilebilen makas ve sinyaller yardımıyla yapıldığı Merkezi Trafik Kontrol (CTC) sisteminde, bölge içindeki hatların işgal durumu, makasların pozisyonu (anahat veya yan hat için düzenlenmiş olduğu) ve sinyallerin görünüşlerine ilişkin bilgiler, kontrol sistemine ait panoda görülebilir. Bu pano üzerinde,

yönetim bölgesi içinde kalan demiryolu hatlarının çizilmiş bir modeli bulunmaktadır. Hat, makas ve sinyal bilgileri pano üzerindeki hat modelinin ilgili yerlerinde bulunan renkli ve ışıklı göstergeler yardımıyla temsil edilir.

Demiryolu trafik yönetimine ilişkin işlemler trafik kontrolörü (dispeçer, tren dispeçeri) sayesinde yapılır. Stratejik kararlar dışında kalan (örneğin çizelgeden tren çıkarma veya çizelgeye tren ekleme vb.), yönetime ilişkin tüm kararları dispeçer verir. Dispeçer, bu kararları verirken, aktif görev yapan personel/donanım'dan sürekli bilgi alır; topladığı diğer bilgilerin de sayesinde gerekli kararları verir ve yine bu personel/donanım aracılığıyla verdiği kararları uygular. Dispeçer verdiği kararlarda, demiryolu trafik yönetimine ilişkin hedefleri, yönetmelikleri, kuralları ve emirleri gözönünde bulundurmak durumundadır.

Her demiryolu işletmesinin trafik yönetimine ilişkin kurallarının başında 'güvenlik' yer alır. Trafik kontrol işlemi, hattın belirli noktalarında bulunan operatörler aracılığıyla yürütüldüğü hatlarda, dispeçerin tren kazalarını önleme konusunda tam sorumluluğu vardır. En büyük hata, operatörlere gönderilen kontrol emirlerinin çatışması; yani, çatışan tren hareketlerine yol açacak emirlerdir (örneğin zıt yönde hareket eden iki trenin komşu iki istasyon arasında seyretmesine izin verilmesi vb.). Diğer önemli bir hata, planda yapılan değişikliğin yolda çalışmakta olan personele bildirilmemesidir. Güvenli bir trafik akışının sağlanabilmesi için, trafiğin sinyal/işaret'lerle yönetildiği bölgelerde bir tren durdurulacaksa, dispeçer doğru kontrol emrini zamanında göndermeli; operatör gerekli sinyal/işaret düzenlemesini yapmalı; ve makinist bu sinyal/işaret'e uymalıdır.

Bir trenin, değişik sebeplerle hareket planına uygun olmayan seyiri, trenin çizelgeden sapmasına neden olmakta; böylece, trenlerarası çatışmaların ortaya çıkmasına neden olmaktadır. Trenlerarası çatışmalar yeteri kadar önceden saptanmalı ve çizelgede gerekli düzenlemeler yapılarak, yeni çizelge hayata geçirilmelidir. Burada, çizelgeden sapmaların saptanması ve çizelgede yapılan değişikliğin uygulanması 'kontrol'; çizelgenin, değişiklik çatışmalar ortadan kaldırılacak biçimde yeniden düzenlenmesi ise, 'yeniden çizelgeleme' işlemlerini oluşturmaktadır. Tren hareketlerinin yeniden çizelgelemesi, belirlenen güvenlik ve etkinlik hedefleri doğrultusunda, trenlerin karşılaşma/önegeçme yer ve zamanlarının belirlenmesi, hangi trenin hangi hatta ve ne kadar süreyle bulunması gerektiğinin saptanması anlamına gelmektedir. Burada etkinlik ölçütü, trenlerin, çizelgeden toplam sapma miktarı, bir

başka deyimle, çatışma gecikmeleri toplamı olmaktadır. Bu gecikmelerin minimizasyonu da amacı oluşturmaktadır.

Görüldüğü gibi, demiryolu trafik yönetimi oldukça zor bir problemdir. Bölgenin herhangi bir yerinde bulunan potansiyel bir çatışmanın çözümünün, bölgenin başka kesimlerinde hareket etmekte olan diğer trenleri de etkileyebileceği gözönüne alındığında, problemin ne kadar karmaşık ve büyük boyutlu olduğu daha iyi anlaşılabilir.

2.3 Demiryolu Trafik Yönetiminin Hiyerarşik Analizi

Trafik yönetimi, belirli bir düzen çerçevesinde yürütülen işlemler bütünüdür. Bu nedenle yapılacak analizin hiyerarşik yapıda olması da normaldir. Demiryolu trafik yönetiminin iki önemli yapısal elemanı Trafik Yönetim Merkezi ve Trafik Yönetim Bölgesidir. Bu iki eleman arasındaki bilgi akışı, trafik yönetiminin gerçekleşmesinde büyük öneme sahiptir. Şekil 2.1'de gösterildiği gibi, bu iki elemanın kendi iç yapıları, birbirleriyle olan ilişkileri ve demiryolu trafik yönetimine olan katılımları, hiyerarşik bir yapıda açıklanmaktadır.

2.3.1 Trafik Yönetim Merkezi

Trafik Yönetim Merkezi, adından da anlaşılacağı gibi, yönetimin gerçekleştiği yerdir. Yönetim işlemi fiziki sınırları belirli bir demiryolu kesimi için yürütülür. Yönetim merkezi binası genellikle bölge içinde bulunur. Merkezde bilgi alışverişini gerçekleştirmek üzere çeşitli haberleşme araçları bulunur. Gelişmiş yönetim merkezlerinde gözlem ve kontrol amacıyla bir trafik kontrol sistemi de yer alır. Bazı bölgelerde iki farklı trafik kontrol sistemi (örneğin, kontrol işleminin hat boyundaki personel aracılığıyla yapıldığı, TMI ve kontrolün bir merkezden uzaktan kumanda edilebilen makaslar ve sinyaller aracılığıyla yapıldığı, CTC), bölgenin değişik kesimlerindeki trafiği kontrol etmek amacıyla kullanılabilir. Trafik yönetimi işi, dispeçer tarafından yapılır. Haberleşme ve kontrol sistemlerini kullanan dispeçer, üst yönetimle bilgi ve karar alışverişi yaptığı gibi, diğer yönetim merkezleri ve ilgili personelle de sürekli iletişim kurmak durumundadır. CTC sistemini kullanan yönetim merkezi ve yönetim bölgelerinde dispeçer, bölge içindeki trafiğin durumuyla (örneğin, her trenin bulunduğu yer)

hattın durumuna ilişkin bilgilere, (örneğin, hat kesimlerinin işgal durumu, makasların pozisyonu ve sinyallerin görünüşü) kontrol sistemini gözleyerek erişebilir. Bölge içinde gerçekleşen trafik hareketlerine ilişkin yol-zaman çizelgesi dispeçer tarafından elle ya da kontrol sistemi tarafından otomatik olarak çizilir. Bu çizelge, bölge içindeki trafiğin geçmişte nasıl bir hareketlilik gösterdiği konusunda bilgiler içerdiği için, gelecekte nasıl bir durum alacağı konusunda da dispeçere bir fikir verir. Böylece, trafik yönetiminin amaçları doğrultusunda gerekli kararların alınabilmesi için destekleyici bir unsur elde edilmiş olur.

2.3.2 Trafik Yönetim Bölgesi

Trafik Yönetim Bölgesi, fiziki sınırları belirli bir demiryolu kesimidir. Bölge içinde bir ya da daha çok paralel hat olabileceği gibi, bir alt şebeke de bulunabilir. Bölgenin komşu yönetim bölgeleriyle olan bağlantısı, genellikle terminal istasyonları aracılığıyla sağlanır. Bölge içinde kalan hatlar boyunca istasyon/sayding'ler bulunur. Trafik yönetimi açısından bu yerler, trenlerin karşılaşma/önegeçme yaptıkları noktalardır. Bölgedeki trafiğin güvenli hareketi, hat boyunca bulunan işaret/sinyal'ler aracılığıyla düzenlenir. Bu düzenleme, trafik kontrol sisteminin izin verdiği kurallar çerçevesinde yapılır. Trafik Yönetim Bölgesi içindeki trafiğin hareketi Trafik Yönetim Merkezi tarafından sürekli izlenir. İki tren arasında bir karşılaşma ya da önegeçme gerçekleştirileceği zaman, trenler bu işlemin yapılacağı yere varmadan önce gerekli hazırlıklar yapılmalıdır. Bu amaçla, dispeçer, öncelikle, çatışmanın çözüleceği istasyon/sayding'in ilgili yan hattının işgal edilmiş olup olmadığını kontrol eder. Eğer hat işgal edilmemişse ilgili makası doğru pozisyona getirir. Daha sonra da, ilgili sinyalin 'ilerle' görünüşünü almasını sağlar. Bu işlemlerin ardından da tüm hazırlıkların tamam olup olmadığını kontrol eder. Dispeçer sorumlu olduğu bölgeyi çok iyi tanımak zorundadır. Özellikle, yüksek eğimli kesimlerin, küçük yarıçaplı kurpların ve hız kısıtlaması uygulanan kesimlerin yerlerini bilmek zorundadır.

Her bölgenin Trafik Yönetim Merkezi kendi bölgesi içindeki demiryolu trafiğinden sorumludur. Trafik, başlangıç ve son noktası bölge içinde olan, bölgesel trafik olabileceği gibi; komşu bir bölgeden gelip, bölgeyi katettikten sonra bir başka komşu bölgeye geçen, bölgeler arası trafik de olabilir. Demiryolu trafiği, hareket halindeki yolcu ve yük trenleri, demiryolu tesislerinin bakım ve onarımını yapan iş katarları ve demiryolu şebekesi içindeki lokomotif ve vagon dağıtımını gerçekleştiren dizilerden oluşur. Bu taşıtların hareketleri plan duyarlı ya da talep duyarlı işletmecilik koşulları altında düzenlenir.

2.4 Demiryolu Trafik Yönetiminin İşlevsel Analizi

Dispeçer, bir mesai süresince çeşitli işlerle ilgilenmek zorundadır. Bu amaç doğrultusunda, “dispeçer şu anda ne yapıyor ?” sorusuna yanıt oluşturan en az sayıdaki açıklayıcı terim araştırılmıştır. Böylece, yapılan işlerin tümünü içeren aşağıdaki işlevler tanımlanabilmiştir:

- Tren hareketlerini izleme ve koordinasyon,
- Kayıt tutma,
- Planlanmamış olayları ele alma,
- Acil durumları ele alma,
- Plan hazırlama, ve
- Tren hareketlerini başlatma ve durdurma.

Tren hareketlerini izleme ve koordinasyon, trafik yönetimi için gerekli bilgilerin elde edilmesi ve ilgili birimler arasında bilgi alışverişinin sağlanmasını sağlar. Dispeçer, trafiğe ilişkin bazı bilgileri CTC panosundan alabildiği gibi, telefon ve telsiz cihazlarını da kullanarak topladığı bilgilerle bölgenin durumunu saptar ve gerekli koordinasyonu sağlar.

Kayıt tutma, tren işletilmesine ilişkin bilgilerin saklanması işlevini yerine getirir. Her demiryolu işletmesinde belirli işler için hazırlanmış formlar bulunmaktadır. Bilgilerin kaydı, bu formlar üzerinde bulunan boşluklar doldurularak yapılır. Ayrıca, dispeçerin telefon ve telsiz görüşmeleri de manyetik bantlara kaydedilerek belirli bir süre saklanır. Kaydı yapılan bilgiler arasında; gönderilen emirlerin içerikleri ve ilgili personelin adları, yer ve zaman bilgileri, trenlerin belirli noktalardaki kalkış-varış zamanları, trenlerdeki boş ve dolu vagon sayıları, tren ağırlıkları, tren personelinin görevde kaldığı süre, bakım ve onarım işleri için ayrılan yol-zaman limitleri, gecikme ve hava durumuna ilişkin raporlar yer alır.

Planlanmamış olayları ele alma, adından da anlaşılacağı gibi, önceden planlanmamış olaylar gerçekleştiğinde yapılması gereken işleri kapsar. Bir ‘planlanmamış olay’, mevcut planların yenilenmesini zorunlu hale getirebilir. Planlanmamış olaylar arasında lokomotif arızası, raydan çıkma, katar kopması, tren personelinin hastalanması, bakım personelinin

başladığı işi bitirebilmek için daha fazla zaman istemesi, kötü hava koşulları ve haberleşmedeki aksaklıklar sayılabilir.

Acil durumlarda planlanmamış olaylar arasındaki temel farklılık, acil durumlarda yaşam ve mülkiyetin tehlike sınırına yaklaşmış olmasıdır. Acil durum, ani olarak oluşur ve dispeçerin zaman yitirmeden olaya müdahale etmesini gerektirir. 'Planlanmamış' olarak sınıflandırılan bir olay, daha sonra acil duruma dönüşebilir. Sel ve yoğun kar yağışı acil durumlar arasında sayılabilir.

Plan hazırlanması mesai öncesinde başlar ve mesai süresince devam eder. Bu işlev, trafiğin kontrolü için zorunlu, diğer işlevlerin yerine getirilmesi için de bir ön koşuldur. Bu işlev içinde tren hareketleriyle ilgili kayıtların gözden geçirilmesi, donanımın kontrol edilmesi, trafiğin durumunun saptanması ve önceden hazırlanmış planların anlaşılması sayılabilir. Planlanmamış ya da acil bir durumla karşılaştığında, dispeçer durumun tanımlanması amacıyla gerekli bilgileri toplar ve bölge içinde doğabilecek potansiyel problemleri saptar. Daha sonra bu problemleri önleyecek gerekli planlamayı yapar. Örneğin, trenlerin gecikmesi ve önceden hazırlanmış çizelgeden sapma göstermeleri sebebiyle doğabilecek potansiyel çatışmaların önlenmesi için yeni bir çizelgenin hazırlanması gereklidir. Dispeçerin günlük işleri arasında oldukça önemli bir yer tutan yeniden çizelgeleme, işletme için belirlenmiş bir etkinlik ölçütü gözönünde bulundurularak yapılır.

Tren hareketlerini başlatma ve durdurma, hazırlanmış hareket planlarının uygulanması, bir başka deyimle trafik kontrol işleminin gerçekleştirilmesi işidir. CTC sisteminin kullanılarak makas ve sinyallerin uzaktan kontrolü, kontrole ilişkin emir ve bilgilerin iletilmesi amacıyla haberleşme donanımının kullanılması bu işlev içinde yer alır.

Yapılan gözlemler bir dispeçerin, yukarıda belirtilen ilk dört işlev için mesai süresinin yaklaşık olarak yüzde 80'ini, plan hazırlama için yüzde 4'ünü, ve tren hareketlerini başlatma ve durdurma için ise yüzde 16'sını harcadığı saptanmıştır.

2.5 Yeniden Çizelgeleme ve Trafik Kontrolü

Bu kısımda dispeçerin, planlanmamış olaylar karşısında, önceden hazırlanmış çizelgeyi yenilemesi (yeniden çizelgeleme) ve tren hareketlerini başlatma ve durdurma (trafik kontrolü) işlevleri daha detaylı olarak ele alınacaktır.

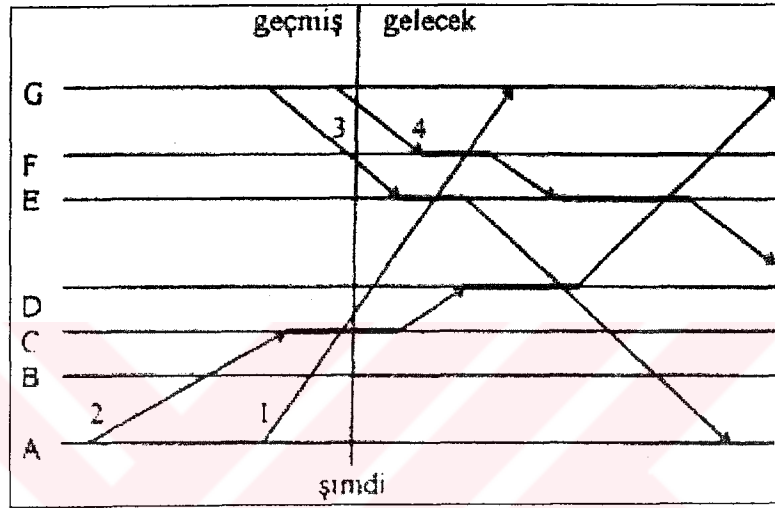
Çizelgeler, her tren için planlanmış bir hareket planıyla birlikte, diğer trenlerle olan karşılaşma/önegeçme'lerin yer ve zamanlarını içerir. Her bölge içindeki trafiğe ait çizelgelerin bir araya gelmesiyle, belirli bir demiryolu hattına ya da şebekesine ait çizelge (taktik trafik planı) oluşturulur. Bu nedenle, bir bölgeye ait çizelgede yapılacak olan değişikliğin, taktik trafik planı gözönünde bulundurularak yapılması gereklidir.

2.5.1 Önceden Hazırlanmış Çizelge ve Gerçekleşen Çizelge

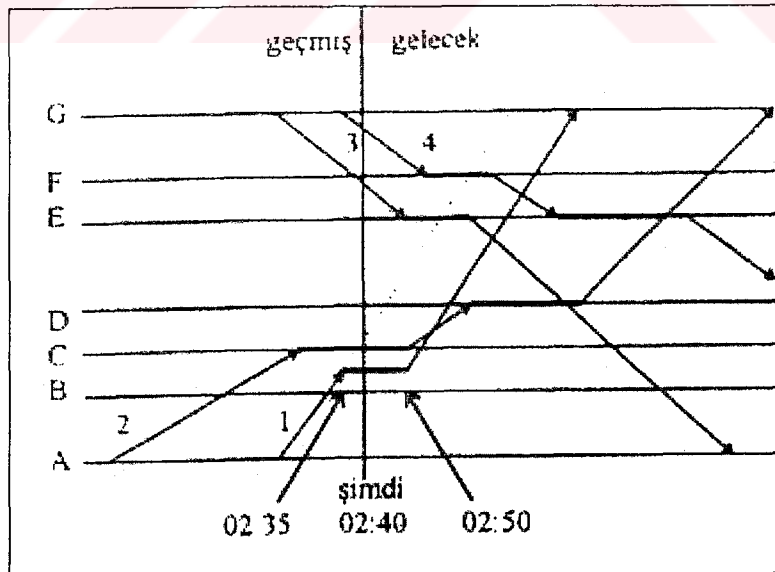
Şekil 2.2, A-G istasyonları arasında kalan tek hatlı bir demiryolu kesiminin önceden hazırlanmış çizelgesini göstermektedir. Burada trenlerin geçmişte gerçekleşmiş hareketleriyle, gelecek için planlanmış hareketlerini görmek mümkündür. Tren çizgilerinin 'şimdi' çizgisiyle kesişimi, trenlerin şu andaki gerçek pozisyonlarını göstermektedir. 'şimdi' çizgisinin solunda yer alan tren çizgileri, yakın geçmişteki tren hareketlerine ait doğru kayıtlardır. Çizginin sağında bulunan tren çizgileri, yakın gelecekte uygulanması öngörülen tren hareketlerini göstermektedir. Şekilden görüldüğü gibi mevcut planda trenler arasında hiçbir çatışma bulunmamaktadır: Tren 1 C'de Tren 2'nin önüne geçmekte; Tren 3 ile E'de, Tren 4 ile de F'de karşılaşmaktadır. Bu önegeçme ve karşılaşmaların tümünde, Tren 1'in öncelikli olarak ilerlemesine izin verilmektedir.

Ancak, demiryolu trafiği zaman içinde değişim gösteren bir yapıya sahiptir. Örnek olarak Şekil 2.3'deki durumu inceleyelim: Yüksek önceliği olan Tren 1'in beklenmedik bir şekilde saat 02:35'te B ve C istasyonları arasında durmak zorunda kaldığını kabul edelim. Saat 02:40'ta trenin makinisti durumu dispeçere bildirir ve saat 02:50'de normal seyrine tekrar başlayabileceğini söyler. Şekilde dispeçerin bu bilgiyi kaydetmesinden hemen sonra, saat 02:40'ta trafiğin durumu da görülmektedir.

Şekil 2.3'ten de görüldüğü gibi, Tren 1 saat 02:50'de normal seyirine başlamasıyla, Tren 2 ile 'izleme ve önegeçme çatışması,' Trenler 3 ve 4 ile de 'karşılaşma çatışması'na girecektir. Bu potansiyel çatışmalar, trenlerin şimdiki durumları dikkate alınarak, işletme ve güvenlik kısıtlarıyla, etkinlik ölçütünün (örneğin, trenlerin planlanmış varış zamanlarından sapmalarının toplamı) ışığı altında çözümlenerek yeni bir çizelge oluşturulur. Yeni bir çizelge, uygulanabilir karşılaşma/önegeçme'lerin hangi trenler arasında, nerede (istasyon veya sayding) ve ne zaman yapılacağına yanıtlarını içermelidir.



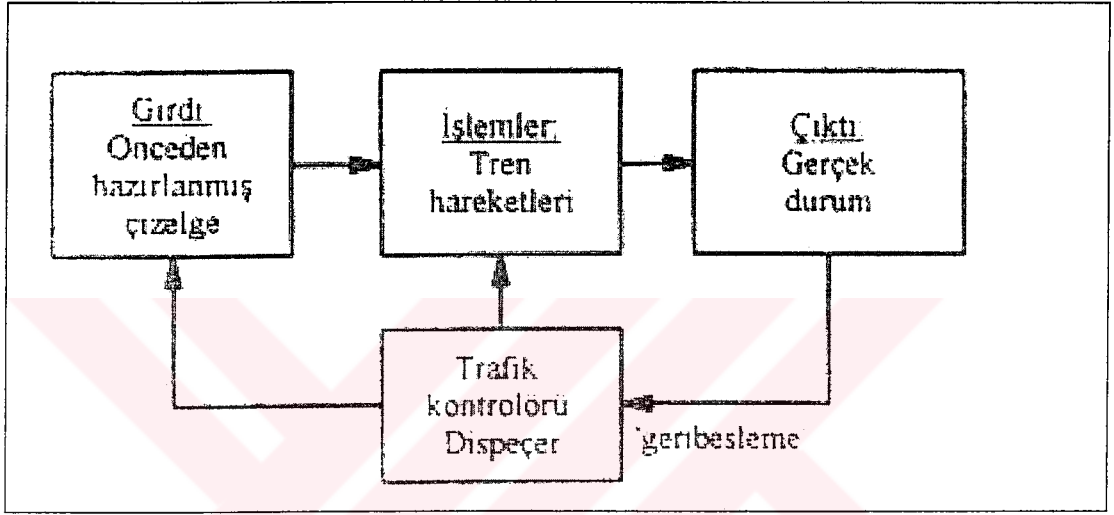
Şekil 2.2 A-G Demiryolu kesimine ait önceden hazırlanmış çizelge



Şekil 2.3 Planlanmamış durum ve ortaya çıkan trenlerarası çatışmalar

Tekrar Şekil 2.3'teki duruma dönecek olursak; Tren 1'in B-C istasyonları arasında planlanmamış duruşuna ilişkin bilgi, trafik yönetim merkezine eriştiği anda, yeniden çizelgeleme ve kontrol işlevleri aktif hale gelir. Tren 1'in saat 02:50'de normal seyirine

devam etmesi durumunda, önceden hazırlanmış çizelgeden 15 dakikalık bir sapma yapacağı görülmektedir. Bu bilginin elde edildikten sonra değerlendirilmesi; bir başka deyimle, Tren 1'in 02:50'de normal seyirine başladıktan sonra, belirli genişlikteki bir zaman aralığı içinde, diğer trenlerle olan ilişkisinin belirlenmesi 'kontrol' işlevinin başlangıç bölümünü oluşturur. Bu işlem, Tren 1'in sistemdeki diğer trenlerle çatışmaya girip girmeyeceğinin belirlenmesi amacıyla yapılır (Şekil 2.4).



Şekil 2.4 Trafik kontrolünde 'geribesleme'

Bu örnekte, Tren 1, Trenler 2, 3 ve 4 ile çatışmaya girmektedir. Yeniden çizelgelemeyle bu çatışmalar çözülerek; yeni çizelgeye uygun olarak, tren hareketlerinin başlatılması/durdurulması gibi kontrol işlemleri yerine getirilir. Görüldüğü gibi kontrol ve yeniden çizelgeleme iç içe bulunmaktadır.

2.5.2 Yeniden Çizelgeleme ve Trafik Kontrolü için Gerekli Bilgiler

Yeniden çizelgeleme ve trafik kontrolü için oldukça yüklü bir bilgi birikimine ihtiyaç vardır. Bu bilgileri şu şekilde sınıflandırmak mümkündür:

- Kurallar ve düzenlemeler,
- yol geometrisi bilgileri,
- hat bilgileri,

- trafik bilgileri,
- trafik kontrol sistemi bilgileri,
- durum bilgileri ve etkinlik ölçütü.

Dispeçer işletmenin yönetimine ilişkin kural ve düzenlemeleri bilmek zorundadır. Bunlar içinde güvenlikle ilgili olanlar da bulunmaktadır. Dispeçer bölgeye ait olan çizelgeyi çok iyi tanımalı ve gerekli bilgiyi hemen okuyabilmelidir. Diğer bilgi kaynaklarının yer ve içeriklerini de bilmek zorundadır.

Dispeçer yönettiği bölgenin geometrik özelliklerini çok iyi bilmelidir. Mümkünse bölgeyi bir lokomotif üzerinde seyahat ederek tanımalıdır. Bölgedeki yol kesimlerinin eğimi, eğriliği, dar ve alçak geçişleri (örneğin yarmalar, tüneller ve altgeçitler) ve bunların yerleri hakkında bilgi sahibi olmak durumundadır. Yolun özel geometriye sahip ve yapısal olarak kötü kesimleri de bu gruba dahildir.

Bölge içinde kalan demiryolu hattının sınırları; hat boyunca yer alan istasyon, sayding, triyaj istasyonu ve kavşak noktalarının yer, sınır ve kapasiteleri; bu kesimler arasında uzanan hat kesimlerinin kullanım tercihleri; izin verilen yasal en yüksek hız ve belirli kesimlerdeki hız kısıtlamaları; hemzemin geçit ve köprülerin yer ve özellikleri; trenlerin durmaması gereken ve telsiz haberleşmesinin sınırlı yapılabildiği hat kesimleri; endüstri kuruluşlarının yer ve özellikleri; ve tren hareketlerini etkileyen diğer unsurlar, hat bilgileri arasında yer alır.

Plan duyarlı bir işletmede demiryolu trafiğinin büyük bir bölümünü, çizelgeye uygun olarak işletilen trenler oluşturur. Her trenin numarası, ilk kalkış ve son varış istasyonları, planlanmış ilk kalkış ve son varış zamanları, ve seyri boyunca belirli noktalara varış ve bu noktalardan kalkış zamanları bilinmelidir. Trenlerin içeriğine ilişkin olarak lokomotif tipi (dizel, elektrikli), uzunluğu, tonajı, temel öncelik sayısı, boş ve dolu vagon sayısı, fren bilgileri, arıza durumu, tehlikeli madde, bozulabilir yük içeriği bilinmesi gerekenler arasındadır. Trenin durumuna ilişkin olarak da yeri, hızı, yönü, seyir bilgileri (belirli noktalar arasındaki tabii ve en kısa seyir süreleri), kalkıştan varışa kadar işgal edeceği hat kesimleri ve istasyon platformları bilinmelidir.

Dispeçer, yönetimindeki tüm trenlerin lokomotiflerinin bazı teknik özelliklerini katar hızlarını tahmin etmek, hızlanma ve yavaşlama yeteneklerini belirlemek, arızalanma olasılığını tahmin etmek için bilmek zorundadır. Dispeçer, bütün demiryolu arabalarının, özellikle aşırı yüksek ve geniş olanların, boyut ve ağırlıklarını bilmelidir. Boyut bilgileri gabari kontrolü için zorunludur. Tren uzunluklarının bilinmesi, sayding uzunluklarıyla uyumun sağlanması açısından önemlidir.

Dispeçer, bölgesinde bulunan sinyalizasyon ve trafik kontrol sisteminin özellikleriyle; bunların kullanımına ilişkin kural ve düzenlemeleri bilmek zorundadır. Kontrol sisteminin izin verdiği en küçük izleme ve güvenlik süreleri her tren çifti ve hat kesimi için bilinmelidir. Dispeçer, hat boyunca yerleştirilmiş olan makas, sinyal, hemzemin geçit ve hareketli köprülerin durumlarını bilmelidir. Hat kesimlerinin işgal durumu, OS'lerin (On Switch; istasyon/sayding'lerin giriş ve çıkışlarındaki makas bölgeleri) işgal edilmesine ait sıralama, bakım ve onarım işleri için ayrılan yol (-zaman) limitleri, geçerli hareket yönü gibi bilgiler, kontrol sistemi aracılığıyla elde edilir.

Dispeçer, hizmetinde olan bütün haberleşme araçlarını tanımalı ve kullanabilmelidir. Kullanılan haberleşme kanallarını çok iyi bilmeli, arıza durumunda ilgili personeli gerekli yere yönlendirebilmeli ve haberleşme problemlerinde yedek kanalları kullanabilmelidir.

Durum bilgileri aracılığıyla bölgenin genel görünümü ve trafiğin etkinliği hakkında bilgi sahibi olunabilir. Herhangi bir anda, bölgedeki trafiğin, hattın, kontrol sisteminin durumuna ait bilgiler bölgenin genel durumunu yansıtmaktadır. Bunlara, özellikle tren personelinin durumuna ilişkin bilgileri de eklemek gerekir; personelin göreve başladığı saat, ilk kalkış ve seyir sırasındaki gecikme, ve mesai süresi personel bilgileri arasında yer alır.

2.5.3 Yeniden Çizelgeleme ve Trafik Kontrolüne Yönelik Bazı Kurallar

Aşağıdaki kurallar demiryolu işletmelerinde yeniden çizelgeleme ve güvenli trafik kontrolünün gerçekleştirilmesi amacıyla uygulanmaktadır. Ancak, bunlar kesin kurallar olmayıp işletmeler arasında farklılık göstermektedir:

- Bir trenin diğeri bir trene göre sınıf, yön veya özel bir hak dolayısıyla önceliğı vardır.
- Özel hak emirle verilir; sınıf ve yön çizelgede belirlidir.
- Özel hak, sınıf ve yöne göre üstünlüğe sahiptir.
- Yön, aynı sınıftaki trenlerin önceliğinin belirlenmesinde kullanılır.
- Birinci sınıf trenler ikinci sınıf, bunlarda üçüncü sınıf trenlere göre önceliklidir.
- Düzenli işleyen trenler ihtiyari trenlere göre önceliklidir.
- Aksi belirtilmedikçe, bir tren ilk hareket istasyonundan önceliğe sahip tüm trenlerin planlanmış varış zamanından ya da planlanmış kalkış zamanından önce kalkmamalıdır.
- Bir tren uygun sinyal verilene kadar kalkmamalıdır.
- Aksi belirtilmedikçe, iki ihtiyari trenin karşılaşması durumunda, üstünlüğü olmayan yönde hareket eden tren saydingte bekletilmelidir.
- Bir buluşma noktasında, üstünlüğü olmayan tren saydingte bekletilmeli ve ana hattı, üstün trenin zamanından en az beş dakika önce boşaltmalıdır.
- Aksi belirtilmedikçe, bir tren planlanmış zamanından önce hareket etmemelidir.
- İki veya çok hatlı kesimlerdeki hatların yönleri tahsis önceden belirlenmiş kurallara göre yapılır.
- Bir karşılaşma için hazırlık yapılırken, eğer pratik ise, buluşma noktasına ilk varan treni yan hatta yönlendirmek, iyi bir buluşma gerçekleştirmek ve zaman kazanmak bakımından önerilebilir.
- Yol düzenlenmesi için hazırlanmadığı durumlarda, makaslar normal pozisyonda ve sinyaller 'dur' görünüşünde tutulmalıdır.
- Makas ve sinyaller, en yakın tren hareketinde gecikme olmaması için, yeteri kadar önceden hazırlanmalı; ve tren, makası geçer geçmez normal pozisyona getirilmelidir.

(Bölüm 2 içinde verilen bilgiler ve şekiller, Şahin (1996) “*Trenlerarası Çatışma Yönetimine Dayalı Trafik Kontrolü İçin Bir Karar Destekleyici Sistem*” adlı kaynaktan alınmıştır.)

3. YAPAY SİNİR AĞLARI

3.1 Giriş

Yapay sinir ağı, insan beyninin temel çalışma prensiplerini esas alarak geliştirilmiş bir algoritma türüdür. Bu algoritma da, tıpkı insan beyni gibi, öğrenme, hatırlama ve genelleme gibi işlemleri yapabilme becerisine sahiptir. Bu nedenle geleneksel çoğu algoritmanın yetersiz kaldığı işlemleri, oldukça hızlı ve doğru bir biçimde yapabilme yetisine sahiptir.

Yapay sinir ağlarının nasıl çalıştığının daha iyi anlaşılabilmesi için, insan beyninin temel çalışma prensiplerinin kavranması gerekmektedir.

3.2 İnsan Beyninin Temel Çalışma Prensipleri

İnsanoğlu, varoluşundan beri, doğadaki sistemin nasıl çalıştığını merak etmiştir. Dolayısıyla, ilk önce kendi organizmasına yönelmiş, duyu ve düşüncelerinin nasıl oluştuğuna bilimsel ve felsefi açıklamalar bulmaya çalışmıştır. Ancak insanoğlu pek çok soruya cevap bulabildiği halde, beyin nasıl çalıştığını keşfedebilmesi yüzyıllar boyunca mümkün olmamıştır.

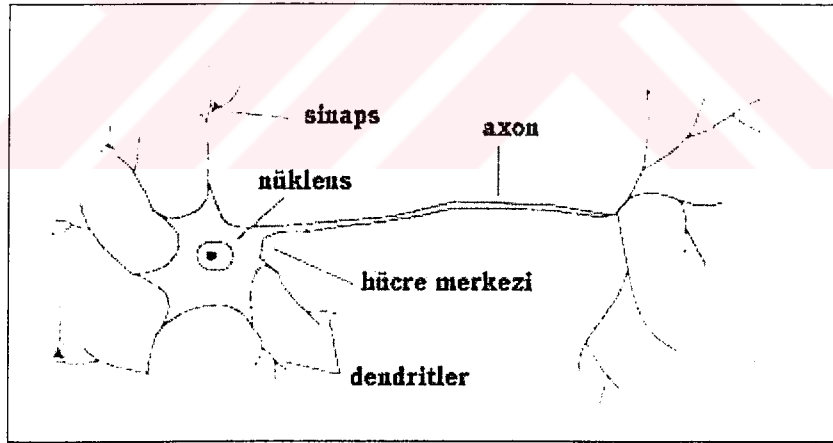
Arkeolojik kazılarda bulunan, tarihöncesi döneme ait insan kafataslarındaki operayon izleri de, daha o çağlarda insanların beyin üzerinde yaptıkları çalışmaların bir kanıtıdır. Ancak bilimsel anlamdaki ilk çalışma, Eski Yunanlı filozof ve matematikçi Aristoteles (M.Ö 384 - 322) tarafından gerçekleştirilmiştir. Yunan ve Roma imparatorluklarının yıkılması sonucu, “Orta Çağ” ya da diğer bir deyişle “Karanlık Çağ”a girilmiş, insanoğlu savaşmayı, bilim ve sanatla uğraşmaya tercih etmiştir. 15. yy.’da İtalya’da başlayan Rönesans sonucu sanata, Martin Luther (1483 - 1546) tarafından Almanya’da başlatılan Reform hareketleri sonucu felsefeye ve 18. yy. İngiltere’inde gerçekleştirilen Sanayi Devrimi sonucunda da teknolojiye verilen önem artmış ve bilimsel çalışmalar ivme kazanmıştır. Günümüzde, insan beyninin nasıl çalıştığı hala tam olarak açıklanamamış olsa da, ortaya konan bazı kanıtlar, bu işlemler zincirinin temellerini gözler önüne sermektedir.

İnsan beyni, yaklaşık olarak 10^{11} adet, nöron adı verilen sinir hücresinden meydana gelir. Görme, işitme, dokunma, koku alma, konuşma gibi temel duyu, beyin farklı bölgelerinde,

çok sayıda nöron tarafından paylaşılarak gerçekleştirilir. Bu nedenle, bazı nöronların ölmesi halinde dahi, bu işlemler diğer nöronlar tarafından gerçekleştirilmeye devam edilebilir.

Şekil 3.1’de görülebileceği üzere, tipik bir nöron, soma, axon ve dendritler olmak üzere 3 ana bölümden oluşur. Soma, hücre çekirdeği olan “nükleus” un bulunduğu ana kısımdır. Dendritler, bir hücreyi, komşu hücrelere bağlayan, elektriksel olarak pasif bağlantılardır. Axon ise, nöronun ürettiği elektrik sinyallerini taşıyan, uzun, silindirik bağlantı elemanıdır. Axonlar, sinaps adı verilen, nöronun ürettiği sinyali, komşu nöronlara ileten dallarla son bulur. İnsan beyinde, nöron başına ortalama 10^4 sinaps bulunur.

Sinir hücreleri, denge halindeyken -85 mV değerinde bir gerilim altındadır. Her hücre için farklı olabilen eşik gerilim değeri aşıldığında, hücre içerisine Na^+ (sodyum) iyonları girişi olur ve hücre uyarıcı yönde etkilenir. Eğer hücreye gelen sinyaller, eşik gerilimini aşamıyorsa, hücre içerisine K^+ (potasyum) iyonları girer ve hücre bastırıcı yönde etkilenir.



Şekil 3.1 Tipik bir sinir hücresi örneği.

3.3 Temel Yapay Sinir Ağı Modelleri ve Öğrenme Kuralları

Şekil 3.2’de, biyolojik nöronun, McCulloch ve Pitts tarafından geliştirilen, basit bir matematiksel modeli görülmektedir. Bu model, çoğunlukla, M-P nöronu olarak adlandırılır. Bu modelde, i . nöron, giriş değerlerinin ağırlıklı toplamını hesaplayarak, toplamın eşik değeri olan θ_i ’nin üzerinde ya da altında olmasına bağlı olarak, 1 (ateşleyen) ya da 0 (ateşlemeyen) çıkış değerlerini verir.

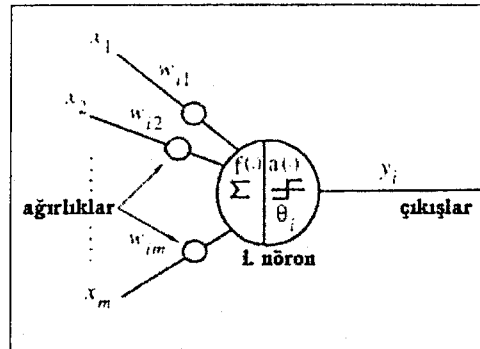
$$y_i(t+1) = \alpha \left(\sum_{j=1}^m w_{ij} x_j(t) - \theta_i \right) \quad (3.1)$$

$$a(f) = \begin{cases} 1 & f \geq 0 \\ 0 & f < 0 \end{cases} \quad (3.2)$$

Burada, $y_i(t+1)$, nöronun $t+1$ anındaki çıkış değerini gösterir. α , öğrenme sabiti, $a(f)$ ise aktivasyon fonksiyonudur. w_{ij} ağırlığı, j nöronunu, i nöronuna bağlayan sinapsın (bağlantının) gücünü simgeler. Pozitif ağırlık değerleri uyarıcı, negatif ağırlık değerleri de bastırıcı sinyalleri simgeler. Eğer, w_{ij} ağırlık değeri 0'a eşitse, bu iki nöron arasında, bağlantının olmadığını gösterir.

Özetle, yapay sinir ağları, aşağıdaki özelliklere sahip, paralel dağılmış, veri işleme yapılarıdır:

1. Nöronlar tarafından oluşturulan bir matematiksel modeldir.
2. Çok sayıda, birbiriyle bağlı işlem elemanı içerir.
3. Bağlantılar (ağırlıklar) bilgiyi taşır.
4. Giriş sinyalleri, işlem elemanına bağlantılar sayesinde ulaşır.
5. Bağlantı ağırlıklarının değerlerinin değiştirilmesi suretiyle, öğrenme, hatırlama ve genelleme yapılabilir.
6. Toplayıcı özelliği, hesaplama gücünü gösterir ve tek bir nöron bütün bilgiyi taşımaz.



Şekil 3.2 McCulloch&Pitts nöronu

Yapay sinir ağı modelleri 3 temel özellikle karakterize edilir; *nöron modelleri*, *sinaptik bağlantıların modelleri* ve yapıları ile ağırlık değerlerini güncelleyecek *eğitme* ya da *öğrenme* kuralları.

3.3.1 Nöronlar

Şekil 3.2’de görüldüğü gibi, bir nöronun veri işleyişi giriş ve çıkış olmak üzere 2 kısımdan oluşur. Giriş bilgilerini birleştirip, *aktivasyonu* sağlamak ve çıkış sinyalini üretmek için bir fonksiyona ihtiyaç vardır. Bu fonksiyon genelde, M-P nöronunda da görülebileceği üzere, lineer bir fonksiyondur.

$$f_i = \sum_{j=1}^m w_{ij} x_j - \theta_i \quad (3.3)$$

Burada, θ_i değeri, i. Nöronun eşik değeridir. Daha karmaşık *aktivasyon fonksiyonları* ise şunlardır:

- Quadratik fonksiyon:

$$f_i = \sum_{j=1}^m w_{ij} x_j^2 - \theta_i \quad (3.4)$$

- Küresel fonksiyon:

$$f_i = \rho^{-2} \sum_{j=1}^m (x_j - w_{ij})^2 - \theta_i \quad (3.5)$$

ρ ve w_{ij} , sırasıyla kürenin yarıçapı ve merkezidir.

- Polinom fonksiyon:

$$f = \sum_{j=1}^m \sum_{k=1}^m w_{ijk} x_j x_k + x_j^{\alpha_j} + x_k^{\alpha_k} - \theta_i \quad (3.6)$$

w_{ijk} , j. ve k. nöronları, i. Nörona bağlayan bağın değeridir ve α_j ve α_k ve ise gerçek katsayılarıdır.

Daha sonra, nöronlar, elde edilen aktivasyon değerini, “a(f)” *aktivasyon* ya da *transfer fonksiyonundan* geçirirler. En çok kullanılan aktivasyon fonksiyonları ise şunlardır:

- Basamak fonksiyonu (Şekil 3.3(a)):

$$a(f) = \begin{cases} 1 & f \geq 0 \\ 0 & f < 0 \end{cases} \quad (3.7)$$

- Hard limit (eşik) fonksiyonu (Şekil 3.3(b)):

$$a(f) = \text{sgn}(f) = \begin{cases} 1 & f \geq 0 \\ -1 & f < 0 \end{cases} \quad (3.8)$$

- Rampa fonksiyonu (Şekil 3.3(c)):

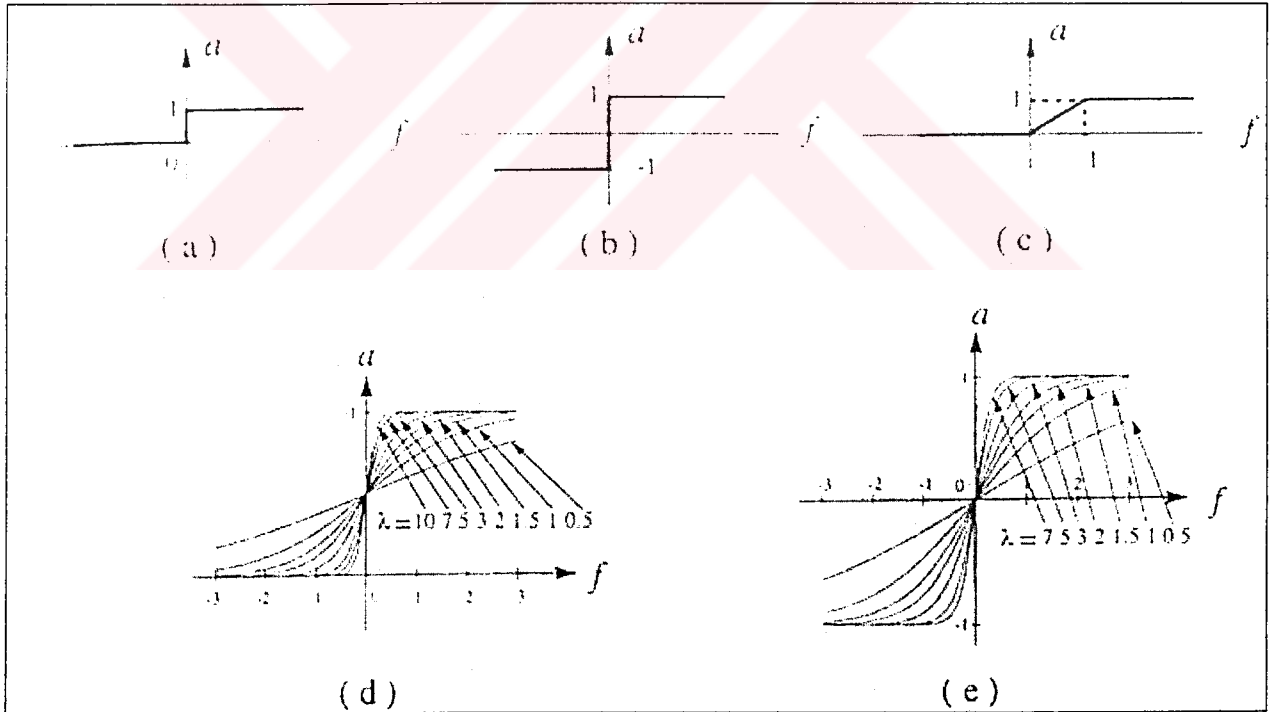
$$a(f) = \begin{cases} 1 & f > 1 \\ f & 0 \leq f \leq 1 \\ 0 & f < 0 \end{cases} \quad (3.9)$$

- Unipolar sigmoid fonksiyonu (Şekil 3.3(d)):

$$a(f) = \frac{1}{1 + e^{-\lambda f}} \quad (3.10)$$

- Bipolar sigmoid fonksiyonu (Şekil 3.3(e)):

$$a(f) = \frac{2}{1 + e^{-\lambda f}} - 1 \quad (3.11)$$



Şekil 3.3 Bazı aktivasyon fonksiyonları. (a) Basamak fonksiyonu. (b) Eşik fonksiyonu. (c) Rampa fonksiyonu. (d) Unipolar sigmoid fonksiyonu. (e) Bipolar sigmoid fonksiyonu

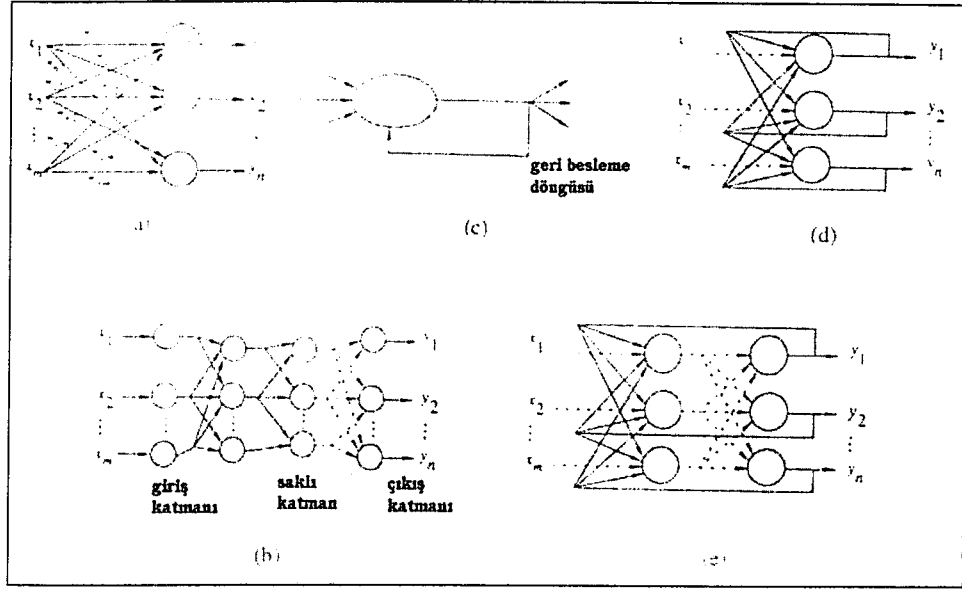
3.3.2 Bağlantılar

Yapay sinir ağları, çok sayıda, birbiriyle bağlı nöronlardan oluşur. Her nöron çıkışı, ağırlıklar (bağlantılar) vasıtasıyla, diğer nöronlara ya da kendisine bağlıdır. Bu yüzden, yapay sinir ağları için, nöronları organize eden yapı ve aralarındaki bağlantı geometrisi bilinmelidir.

Ayrıca, yapay sinir ağlarındaki nöronların gerçekleştirdiği fonksiyona ilave olarak, her bağlantının nerede başladığı ve nerede bittiği de belirtilmelidir. Şekil 3.4'te 5 basit bağlantı geometrisi türü görülmektedir.

Bir katman yaratmak için, ilk önce bir nöron alıp, diğer nöronlarla birleştirmek gerekir. Giriş değerleri, bu nöronlara farklı ağırlıklarla bağlanarak, nöron başına 1 tane olmak üzere, farklı çıkış değerleri üretirler. Bu yapı, Şekil 3.4(a)'da görülen tek katmanlı ileri beslemeli ağ yapısıdır. Bu yapıya daha fazla katman ilave ederek, Şekil 3.4(b)'de görülen, çok katmanlı ileri beslemeli ağ yapısı oluşturulur. Giriş değerlerinin işlendiği katmana giriş katmanı adı verilir ve genellikle sadece giriş sinyalini işler. Ağın çıkış değerleri, çıkış katmanında üretilir. Giriş ve çıkış katmanı arasında bulunan her katman, saklı katman olarak adlandırılır. Bu katman sadece ağın içinde işlem yapar ve dış çevreyle herhangi bir bağlantısı yoktur. Bir yapay sinir ağında, sıfırdan bir kaç adete kadar saklı katman bulunabilir. Bir katmandaki her çıkış değeri, bir sonraki katmandaki her nörona bağlanıyorsa, bu tür ağlara tam bağlı ağlar denir. Şekil 3.4(b)'deki ağ yapısının tam bağlı olmadığı rahatlıkla görülebilir.

Yukarıda belirtilen iki ağ türü (tek katmanlı ve çok katmanlı), ileri beslemeli ağlardır çünkü hiçbir nöron çıkışı, aynı katmandaki ya da bir önceki katmandaki diğer bir nörona bağlantı içermez. Bir nöronun çıkışını, kendisinde ya da bir önceki katmanda giriş değeri olarak kullanan ağ türleri, geri beslemeli ağlar olarak adlandırılır. Kapalı sarmallar içeren geri beslemeli ağlar, kendi kendini besleyen ağlar olarak adlandırılır. Şekil 3.4(c)'de en basit kendini besleyen ağ tipi, kendi kendini besleyen nöron görülmektedir. Tek katmanlı, geri beslemeli bir ağ türü de, Şekil 3.4(d)'de görülebilir. Burada, nöron çıkışı, kendisine, diğer bir nörona, ya da her ikisine birden geri yönlendirilebilir. Şekil 3.4(e)'de ise, çok katmanlı geri beslemeli bir ağ türü görülmektedir. Burada iki tür bağlantı bulunur. Yakın bağlantılar, yakın nöronların bağlantısından, uzak bağlantılar da daha uzak nöronların bağlantısından oluşur.



Şekil 3.4 Basit bağlantı geometrileri. (a) Tek katmanlı ileri beslemeli ağ. (b) Çok katmanlı ileri beslemeli ağ. (c) Kendi kendini besleyen nöron. (d) Tek katmanlı geri beslemeli ağ. (e) Çok katmanlı geri beslemeli ağ.

3.3.3 Öğrenme Kuralları

Yapay sinir ağlarını belirleyen üçüncü önemli unsur, öğrenme kurallarıdır. Matematik gösterimi basitleştirmek için, yapay sinir ağında n adet nöron olduğu ve her nöronun m adet bağlantıya sahip olduğu varsayılmıştır. Ağırlık matrisi;

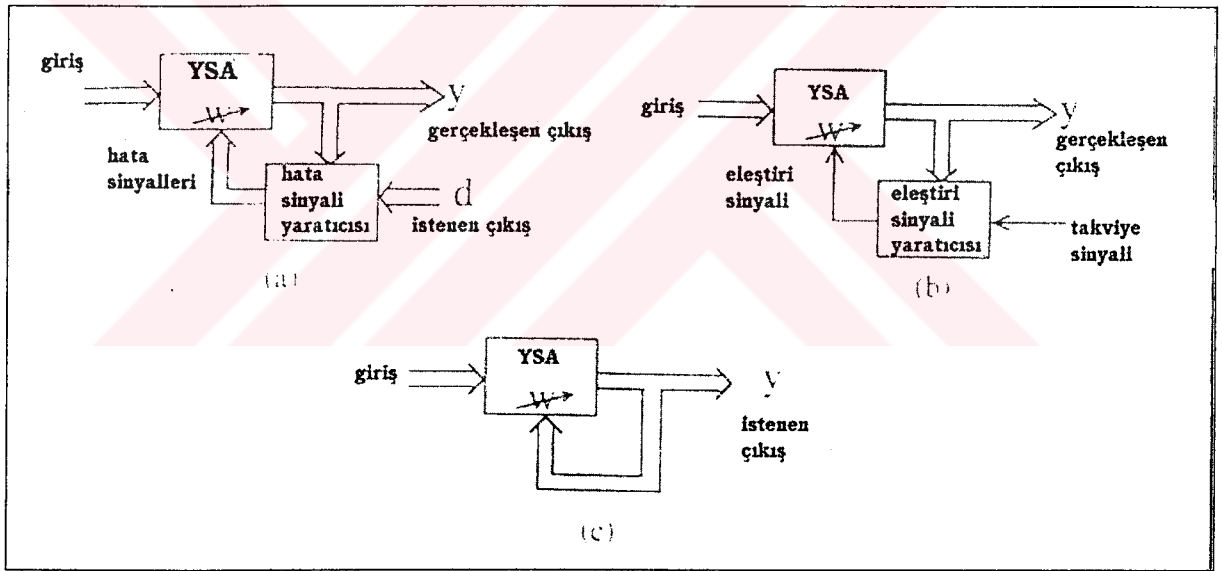
$$w = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_n^T \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix} \quad (3.12)$$

olarak tanımlanır. Burada $w_i = (w_{i1}, w_{i2}, \dots, w_{im})^T$, $i = 1, 2, \dots, n$ nöron i 'nin ağırlık vektörüdür ve w_{ij} de nöron j 'den, nöron i 'ye olan bağlantının ağırlığıdır. Yapay sinir ağının istenilen performansla çalışması için, bu ağırlık matrisi elemanlarının en uygun değerleri almaları gerekmektedir. Yapay sinir ağında ağırlıkların güncellenmesi için uygun bir öğrenme kuralı seçilmelidir. Öğrenme kuralları, *eğiticili*, *takviyeli* ve *eğiticisiz* olmak üzere 3 türdür.

Eğiticili öğrenmede, yapay sinir ağına uygulanan her giriş değeri için, sistemden istenen “d” çıkış değeri de verilmelidir. Bu yolla, ağa çıkış değeri olarak ne vermesi gerektiği öğretilir. Diğer bir deyişle, eğiticili öğrenme türünde, yapay sinir ağı $(x^{(1)}, d^{(1)})$, $(x^{(2)}, d^{(2)})$, ..., $(x^{(k)}, d^{(k)})$ gibi istenen giriş-çıkış çiftleriyle beslenir. Yapay sinir ağına girilen her $x^{(k)}$ giriş değeri için, $d^{(k)}$ istenen çıkış değeri de girilir. Şekil 3.5(a)'da görüldüğü üzere, gerçekleşen çıkış $y^{(k)}$ ve

istenen çıkış $d^{(k)}$ arasındaki fark, hata sinyali yaratıcısında ölçülür ve gerçekleşen çıkış, istenen çıkış değerine yaklaşacak şekilde ağırlık değerlerini düzeltecek hata sinyali üretilir.

Takviyeli öğrenmede, her giriş paterni için, doğru "hedef" çıkış değerlerinin bulunduğu varsayılır. Ancak bazı durumlarda daha az bilgiye sahip olunur. Örneğin, yapay sinir ağına sadece gerçekleşen çıkış değerlerinin "çok büyük" veya "%50 doğru" olduğu söylenebilir. Çok olağan dışı durumlarda, çıkışın doğru ya da yanlış olduğunu belirten çok sınırlı bir geri besleme bilgisi de olabilir. Bu tür eleştirel bilgilerle yapılan öğrenmeye, "takviyeli öğrenme" ve geri besleme bilgisine de "takviye sinyali" denir. Şekil 3.5(b)'de görülebileceği üzere, takviyeli öğrenme, eğitici öğrenmenin bir çeşitidir, çünkü ağ, çevreden bazı geri besleme bilgileri almaktadır. Fakat, geri besleme (örneğin takviye sinyali), yönlendirici olmaktan çok, eleştireldir. Elde edilen çıkışın ne kadar iyi ya da ne kadar kötü olduğu söylenir ve gerçek cevabın ne olduğuna dair bir ipucu bulunmaz.



Şekil 3.5 Üç öğrenme türü. (a) Eğitici öğrenme. (b) Takviyeli öğrenme. (c) Eğitici öğrenme.

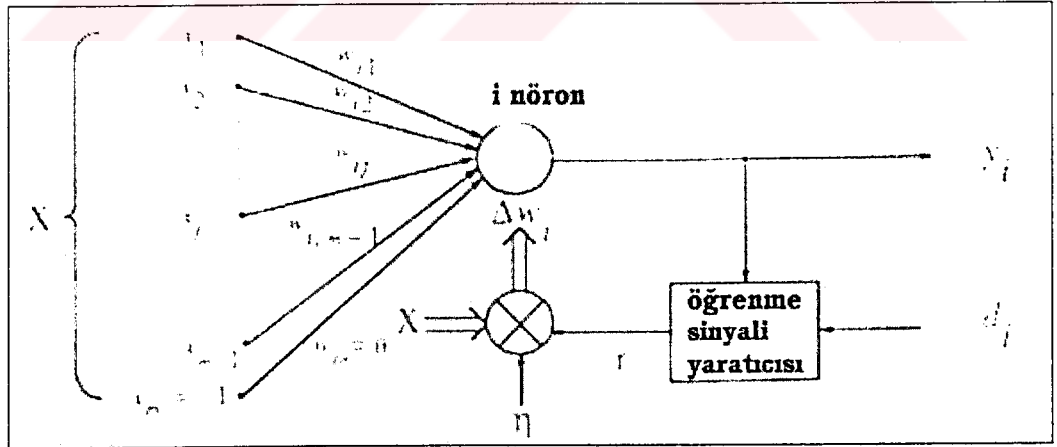
Eğitici öğrenmede, herhangi bir öğretici ya da geribesleme bilgisi bulunmaz (Şekil 3.5(c)). Çıkış değerlerinin ne olması gerektiği ya da doğru olup olmadığı hakkında çevreden gelen hiçbir geri besleme bilgisi yoktur. Ağ, kendi paternlerinin özelliklerini, kurallarını, korelasyonlarını ya da sınıflarını giriş verisinden alarak, çıkış için kendisi kodlamalıdır. Bu özellikleri keşfederken, ağ parametrelerinde değişikliklere gider. Bu işleme kendi kendine organize olma denir. Buna tipik bir örnek, nesnelere gerçek sınıfları hakkında hiçbir bilgi verilmeden, eğitici olarak sınıflandırılmasıdır. Uygun kümeler, nesnelere arasındaki benzerlikler ve farklılıklar keşfedilerek yapılır.

Şekil 3.5'te görülen üç temel öğrenme tipi esas alınarak, Şekil 3.6'da görülen yapay sinir ağındaki bir nöron için genel bir eğitme yapısı oluşturulabilir. Bu şekilde giriş, x_j , $j = 1, 2, \dots, m$ başka bir nöronun çıkışı ya da çevreden gelen bir bilgi olabilir. Eşik parametresi olan θ , bias olarak da adlandırılır. Bias için girişlerden birinin, örneğin x_m , değeri -1 alınarak, ağırlık değerlerinden biri olarak öğrenmeye katılabilir. İstenen çıkış sinyali olan d_i , sadece eğitici öğrenme türünde ya da takviyeli öğrenmede, takviye sinyali olarak mevcuttur. Bu iki öğrenme türünde, i . nöronun ağırlıkları, aldığı giriş sinyallerine, çıkış değerine ve ilgili öğrenme türüne bağlı olarak değiştirilir. Ancak, eğitici öğrenme türünde nöron, ağırlıklarını sadece giriş ve/veya çıkış değerleri sayesinde değiştirir.

t anında yapay sinir ağındaki ağırlık vektörü w_i 'deki değişim, öğrenme sinyali ve giriş $x(t)$ 'ye bağlı olarak şöyle verilir:

$$\Delta w_i(t) = \eta r X(t) \quad (3.13)$$

Burada, η , öğrenme oranını belirleyen, *öğrenme sabiti* adı verilen pozitif bir sayıdır; r ise, w_i , x ve eğer mevcutsa, d_i 'nin bir fonksiyonudur ve *öğrenme sinyali* olarak adlandırılır.



Şekil 3.6 Genel öğrenme yapısı (eğitici öğrenmede d_i bulunmaz)

3.3.3.1 Hebb Öğrenme Kuralı

Bir A hücresinin axonu, bir B hücresini uyarmaya yetecek kadar yakın ise ve ısrarlı ve tekrarlı bir şekilde onu ateşlemede rol oynuyorsa, B'yi ateşleyen hücrelerden biri olarak, A'nın etki oranını arttıracak şekilde bağlantı ağırlıklarının değişimde rol oynayabilir.

Hebb öğrenme kuralında, öğrenme sinyali;

$$r = \alpha(w_i^T x) = y_i \quad (3.14)$$

dir. Burada α , nöronun aktivasyon fonsiyonudur. Hebb öğrenme kuralında, r öğrenme sinyali, nöronun çıkış değeridir. Bu değer, (3.13) formülünde yerine konursa, ağırlık vektörünün artış değeri,

$$\Delta w_i = \eta y_i X \quad (3.15)$$

olur. Böylelikle, ağırlık vektörünün elemanları,

$$\Delta w_i = \eta y_i x_j, i = 1, 2, \dots, n; j = 1, 2, \dots, m \quad (3.16)$$

kadar değiştirilir.

Görüldüğü üzere, Hebb öğrenme kuralı, ileri beslemeli ağlar için eğiticiyiz bir öğrenme kuralıdır. Ağırlık değerlerini güncellemek için, öğrenme sinyali yaratacak istenen çıkış değeri bulunmaz. Bu öğrenme kuralında, eğitmeden önce ağırlıkların sıfıra yakın değerlerden rastgele olarak başlatılması gerekir. Formül (3.15) gösterir ki, eğer giriş-çıkış korelasyon değeri pozitifse, ağırlık w_{ij} artar, aksi halde azalır.

3.4 Tek Katmanlı İleri Beslemeli Ağlar

Bu tür ağlarda, öğrenme kuralı ağırlık değerlerini tahmin edilmesi ve daha sonra amaç fonksiyonuna göre bu ağırlık değerlerinin güncellenmesi sayesinde yapılır. Belli bir sayıda adım sonucunda, optimum ya da optimuma yakın bir sonuca ulaşılır.

$$y_i^{(k)} = a(w_i^T x^{(k)}) = a\left(\sum_{j=1}^m w_{ij} x_j^{(k)}\right) = d_i^{(k)}, i = 1, 2, \dots, n; k = 1, 2, \dots, p \quad (3.17)$$

Burada, $d^{(k)}$, $x^{(k)}$ paterni için istenen cevaptır. Öğrenme işlemi sonucunda, $y^{(k)}$, elde edilen (gerçekleşen) çıkış değerinin, $d^{(k)}$ ya eşit olması istenir. $w_i^T = [w_{i1}, w_{i2}, \dots, w_{im}]$, i . nöronun ağırlık vektörü, $a()$ ise aktivasyon fonsiyonudur.

İlerki bölümlerde, öncelikle lineer eşik birimlerine sahip basit perseptronlar ve perseptron öğrenme kuralı, daha sonra da lineer gradyanlı birimler ve Widrow-Hoff öğrenme kuralı işlenecektir.

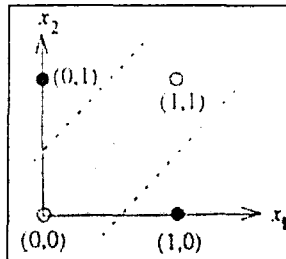
3.4.1 Perseptron Öğrenme Kuralı

Lineer eşik birimlerine sahip basit perseptronlarda, istenen çıkış $d_i^{(k)}$, sadece ± 1 değerini alabilir. Böylece, (3.17) formülü şu hali alır:

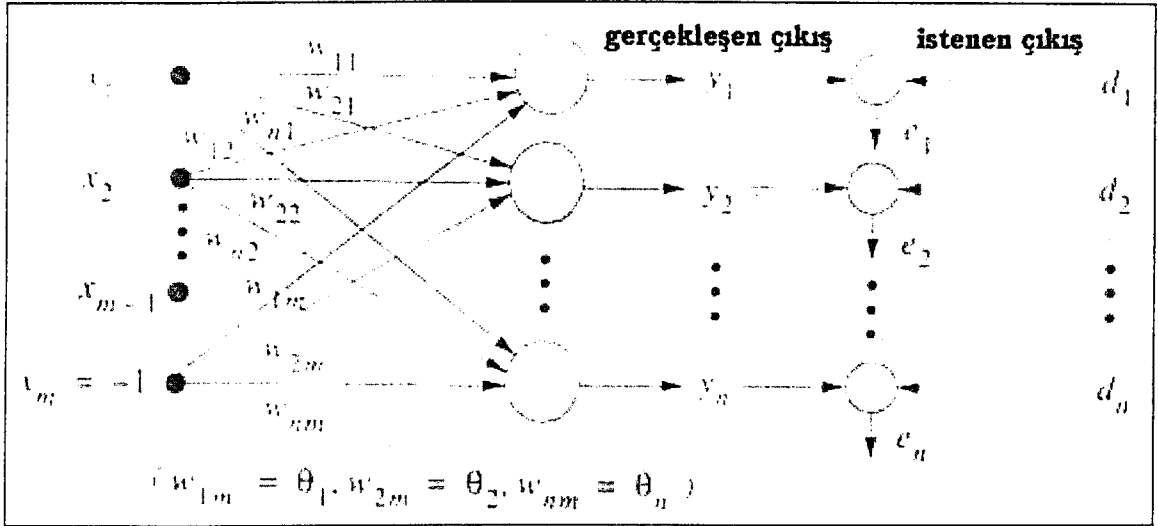
$$y_i^{(k)} = \text{sgn}(w_i^T x^{(k)}) = d_i^{(k)}, i = 1, 2, \dots, n; k = 1, 2, \dots, p \quad (3.18)$$

Burada, i . nöronun ağırlık vektörü olan w_i öyle seçilmelidir ki, $x^{(k)}$ paterni, $d_i^{(k)}$ ile aynı çıkış değerini verebilsin. w_i 'nin pozitif ve negatif izdüşümleri arasındaki sınır, $w_i^T x^{(k)} = 0$ hiperdüzlemi (karar düzlemi olarak adlandırılır) olduğu için, (3.18) formülünde istenen, pozitif ve negatif çıkış değerleri verecek giriş değerlerini ayırabilecek bir hiperdüzlem bulunmasıdır.

Yukarıda belirtilen karar düzlemi, verilen giriş paternleri için her zaman bulunmayabilir. Buna tipik bir örnek, meşhur XOR problemidir. Şekil 3.7 ve Çizelge 3.1'de görüldüğü gibi, girişlerden sadece bir tanesi 1 değerini aldığı anda, istenen çıkış değeri +1, giriş değerlerinin her ikisi de +1 ya da 0 değerini aldığı anda, istenen çıkış değeri -1'dir. Şekil 3.7'de açıkça görülmektedir ki, bu paternleri iki sınıfa ayırabilecek bir düzlem (çizgi) bulunmamaktadır. Bu yüzden, XOR fonksiyonunu basit perseptronla modellemek mümkün değildir. Patern sınıflandırmanın, lineer eşik birimine sahip basit perseptronlarla çözülüp, çözülemeyeceği, lineer olarak ayrılıp, ayrılamadığına bağlıdır. Lineer ayrılabilen problem (sınıflandırma problemi), istenen çıkış değeri -1'e sahip giriş paternlerini, istenen çıkış değeri +1'e sahip giriş paternlerinden ayırabilecek bir karar düzlemine sahip problemlerdir. Eğer birden fazla çıkış nöronu varsa, her çıkış için, bu tür bir düzlem bulunabilmelidir. Başka bir deyişle, eğer bir problem (sınıflandırma), Şekil 3.8'de görülen, basit bir perseptronla lineer olarak ayrılabilirse, şu özelliklere sahip $w_i, i = 1, 2, \dots, n$ ağırlık vektörü bulunabilmelidir:



Şekil 3.7 XOR problemi



Şekil 3.8 Basit bir perseptron

Çizelge 3.1 XOR problemi için çıkış değerleri

x_1	x_2	Çıkış
1	1	0
1	0	1
0	1	1
0	0	0

$w_i^T X > 0$, istenen çıkış değeri = +1'e sahip her x değeri için,

(3.19)

$w_i^T X < 0$, istenen çıkış değeri = -1'e sahip her x değeri için,

Bir karar düzlemi bulabilmek için, perseptronun uygun ağırlık değerlerini bulacak bir öğrenme kuralına ihtiyaç vardır. Eğitim verisini teker teker işleyen ve ağırlık değerlerini ayarlayan öğrenme kuralı, perseptron öğrenme kuralı olarak adlandırılır. Perseptron öğrenme kuralında, öğrenme sinyali, istenen ve gerçekleşen nöron çıkışları arasındaki farktır.

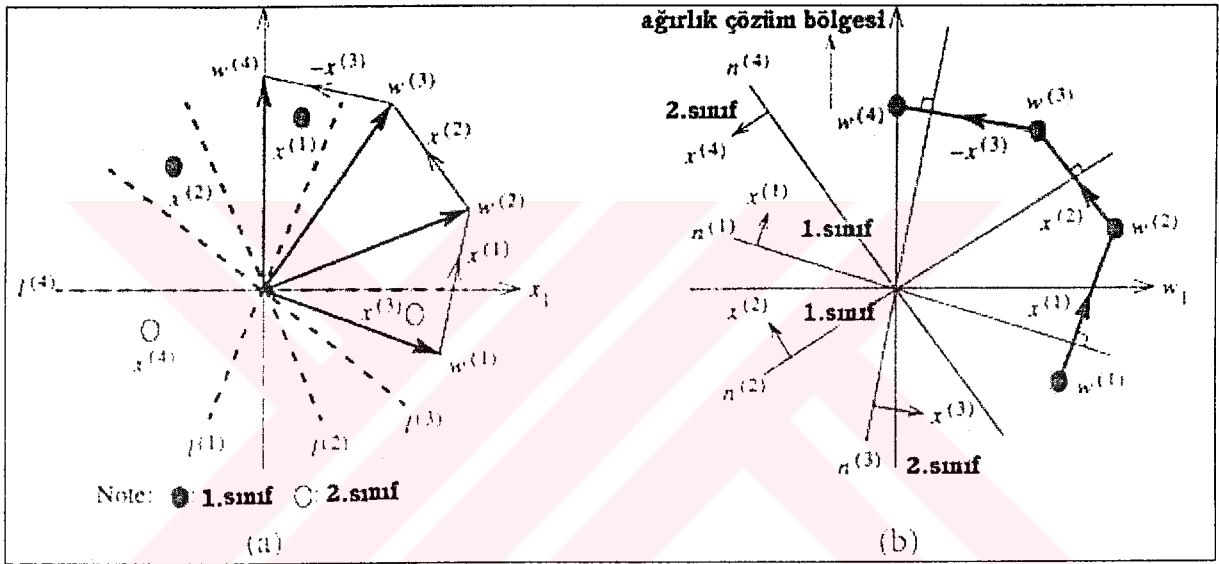
$$r = d_i - y_i$$

(3.20)

Burada $y_i = \text{sgn}(w_i^T x)$ ve d_i de i . nöronun istenen çıkış değeridir. İstenen çıkış değeri, $d_i, \pm 1$ değerlerini aldığına göre, formül (3.13) şu şekilde yazılabilir:

$$\Delta w_{ij} = \eta [d_i - \text{sgn}(w_i^T x)] x_j = \begin{cases} 2\eta d_i x_j & y_i \neq d_i \\ 0 & y_i = d_i \end{cases} \quad j = 1, 2, \dots, m \quad (3.21)$$

Görüldüğü gibi, ağırlık değerlerindeki değişim, gerçekleşen çıkış, y_i , istenen çıkış, d_i 'den farklı olduğunda gerçekleşmektedir. Bu yöntemde ağırlık değerleri istenilen değerden başlatılabilir.



Şekil 3.9 Perseptron öğrenme kuralında ağırlık güncellemeleri

Perseptron öğrenme kuralı ile ağırlık güncellemelerine bir örnek Şekil 3.9'da verilmiştir. Burada, öğrenme sabiti, η , 0.5 alınmıştır. Ağırlık vektörünün aldığı değerler, $w^{(1)}$, $w^{(2)}$, $w^{(3)}$, $w^{(4)}$ olarak gösterilmiştir. İki sınıftan, toplam dört adet giriş paterni vardır; $x^{(1)}$ ve $x^{(2)}$, birinci sınıfa, $x^{(3)}$ ve $x^{(4)}$, ikinci sınıfa aittir. Birinci sınıfa ait paternler için, istenen çıkış değeri + 1, ikinci sınıfa ait paternler için ise - 1'dir. Giriş paternleri, perseptron öğrenme kuralına, $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, $x^{(4)}$ sırasında verilmiştir. $\text{sgn}((w^*)^T x^{(1)}) = \text{sgn}((w^*)^T x^{(2)}) = +1$ ve $\text{sgn}((w^*)^T x^{(3)}) = \text{sgn}((w^*)^T x^{(4)}) = -1$ değerlerini elde edecek w^* , ağırlık vektörü bulunmaya çalışılmaktadır.

$i = 1, 2, 3, 4$ nin bulunduğu ağırlık uzayı görülmektedir. Bu iki seçenekli sınıflandırma problemi için uygun çözümleri, taralı alandaki ağırlık değerlerinin sağladığı görülmüştür.

3.4.2 Adaline

Tek lineer birime sahip ağlara, Adaline (*Adaptive Linear Element*, adaptif doğrusal eleman) denir. Adaline'da, giriş-çıkış ilişkisi doğrusaldır. Çözülecek problem yine Formül (3.17)'deki gibi bir eğitici öğrenme problemidir. Hedef, verilen p adet eğitime paterni kümesi için, $\{(x^{(1)}, d^{(1)}), (x^{(2)}, d^{(2)}), \dots, (x^{(p)}, d^{(p)})\}$, doğru w_i ağırlık setini bulmaktır.

$$\sum_{j=1}^m w_j x_j^{(k)} = d^{(k)}, k = 1, 2, \dots, p \quad (3.22)$$

Bu formül, aslında giriş paternleri $x^{(1)}, x^{(2)}, \dots, x^{(p)}$ lineer bağımsız ise, w^* çözümüne sahip, lineer formüllerin kombinasyonudur. Her biri m boyutlu vektör olan p giriş paterni seti, sadece eğer $p \leq m$ olursa lineer bağımsız olabilir ve Adaline'da en fazla m adet birbirine bağlı ilişki bulunabilir. Lineer birimler için lineer bağımsızlık durumu, eşik birimleri için lineer ayrılabilirlik koşuluyla uyumludur. Lineer bağımsızlık, lineer ayrılabilirliği sağlar ancak tersi doğru değildir.

Formül (3.22)'deki ağırlıkları bulabilmek için, sistemin performansını hesaplayan bir $E(w)$, değer fonksiyonuna ihtiyaç vardır;

$$E(w) = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - y^{(k)})^2 = \frac{1}{2} \sum_{k=1}^p (d^{(k)} - w^T x^{(k)})^2 = \frac{1}{2} \sum_{k=1}^p \left(d^{(k)} - \sum_{j=1}^m w_j x_j^{(k)} \right)^2 \quad (3.23)$$

$E(w)$ küçüldükçe, w_j 'nin değeri iyileşecektir. $E(w)$ normalde pozitiftir ancak $y^{(k)}, d^{(k)}$ 'ya, $k = 1, 2, \dots, p$, yaklaştıkça sifira yaklaşır. Bu yüzden, karesel hata $E(w)$ 'yu minimize edecek ağırlık değerleri bulunmalıdır. Analitik çözüm elde etmek yerine, ağırlıkları rastgele değerlerden başlatıp, sürekli olarak iyileştirebilecek bir öğrenme kuralına ihtiyaç vardır. Formül (3.23)'te verilen değer fonksiyonuyla, w_j ağırlık setini, ağırlık düzlemi boyunca aşağıya kaydırarak iyileştirme gerçekleştirilebilir. Başka bir deyişle, eğim düşümü (gradyan azaltma) algoritması, her w_i ağırlık değerini, $E(w)$ 'nin gradyanına negatif yönde Δw_i kadar değiştirir:

$$\Delta w = -\eta \nabla_w E(w) \quad (3.24)$$

Böylelikle:

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = \eta \sum_{k=1}^p (d^{(k)} - w^T x^{(k)}) x_j^{(k)} \quad (3.25)$$

olur. Eğer bu değişiklikler, her $x^{(k)}$ giriş paterni için ayrı ayrı yapılırsa, $x^{(k)}$ paternine verilen cevaptaki değişim;

$$\Delta w_j = \eta (d^{(k)} - w^T x^{(k)}) x_j^{(k)} \quad (3.26)$$

olarak gerçekleşir. Formül (3.26)'da görülen öğrenme kuralı Adaline öğrenme kuralı ya da Widrow-Hoff öğrenme kuralı olarak adlandırılır. Ayrıca, en küçük kareler yöntemi olarak da bilinir. Bu yöntemde ağırlıklar rastgele değerlerden başlatılır. Formül (3.13)'te görülen genel ağırlık öğrenme kuralından farkı, öğrenme sinyali r , $r = d - y = d - w^T x$ olarak verilmiştir.

Widrow-Hoff öğrenme kuralı, formül (3.21)'deki perseptron öğrenme kuralına çok benzer. Aralarındaki en büyük fark, perseptron öğrenme kuralı, genel Hebb öğrenme kuralından yaratılmış, Widrow-Hoff öğrenme kuralı ise kolaylıkla birden fazla katmana uygulanabilen eğitim düşümü metodundan elde edilmiştir. Ayrıca, perseptron öğrenme kuralı sonlu sayıda öğrenme adımından sonra durur ancak eğitim düşümü yöntemi, en uygun çözüme ulaşınca ya da ağırlıkların değiştirilmesine devam eder. Formül (3.23)'te verilen değer fonksiyonunda ağırlıklar quadratik oldukları için, paternleri ayıran altdüzlem, minimumu (çözümü) $E(w) = 0$ olan parabolik ya da hiperbolik bir eğridir. Bu yüzden, patern altdüzleminde, eğitim düşümü yöntemi, η yeteri kadar küçük olduğu sürece hatayı azaltır. Böylece, herhangi bir başlangıç noktasından yeteri kadar iterasyonla eğrinin minimum noktasına ulaşılarak, istenen çözüm sağlanmış olur.

Lineer nöronlar, Adaline'in ve lineer ağların temel birimleridir. Eğitim düşümü kuralını lineer nöronlu basit perseptronlardan, aktivasyon fonksiyonları değişim gösterebilen lineer gradyanlı nöronlara adapte etmek gerekir. Bu durumda, formül (3.23)'teki değer fonksiyonu,

$$E(w) = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^n (d_i^{(k)} - y_i^{(k)})^2 = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^n [d_i^{(k)} - a(w_i^T x^{(k)})]^2 = \frac{1}{2} \sum_{k=1}^p \sum_{i=1}^n \left[d_i^{(k)} - a \left(\sum_{j=1}^m w_{ij}^T x_j^{(k)} \right) \right]^2 \quad (3.27)$$

halini alır. Burada k , eğitim paternini, p ise toplam eğitim verisi sayısını göstermektedir.

Formül (3.24)'teki eğim düşümü algoritması, formül (3.27)'ye uygulandığında,

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{k=1}^p [d_i^{(k)} - a(\text{net}_i^{(k)})] a'(\text{net}_i^{(k)}) x_j^{(k)} \quad (3.28)$$

elde edilir. Burada $\text{net}_i^{(k)} = w_i^T x^{(k)}$, i. nörona net giriştir ve $a'(\text{net}_i^{(k)}) = \partial a(\text{net}_i^{(k)}) / \partial \text{net}_i^{(k)}$ 'dir. Yukarıdaki formül, ağırlık güncellemesine uygulandığında, gerçek gradyanın, eğitme setindeki tüm paternlerin toplamını içerdiği görülür. Ancak, çoğu durumda, gradyan tahminleri bağımsız örneklerden yapılır. Gerçek gradyanlardan ziyade, gradyan tahminlerini kullanmak için, öğrenme sabiti η 'nın küçük seçilmesi gerekir. k. patern girildikten sonra w_{ij} 'ye uygulanan eğim düşümü,

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \eta [d_i^{(k)} - a(\text{net}_i^{(k)})] a'(\text{net}_i^{(k)}) x_j^{(k)} \quad (3.29)$$

şeklinde olur. Formül (3.29), genellikle *delta öğrenme kuralı* olarak adlandırılır. Bu durumda, formül (3.13)'teki genel öğrenme kuralından, r öğrenme sinyali,

$$r = [d_i - a(w_i^T x)] a'(w_i^T x) \quad (3.30)$$

elde edilir.

Basit perseptronlarda, lineer olmayan aktivasyon fonksiyonları kullanmanın iki ana avantajı vardır. Birincisi, çıkış değerlerini sabit sınırlar içerisinde tutarlar (örneğin, bipolar sigmoid fonksiyonu için, ± 1). Bu, ileri beslemeli ağ yapısının makul sayıda katmana sahip olmasını sağlar. İkincisi, ağırlara lineer olmamayı getirirler, bu da lineer birimlerle çözülemeyen problemlerin çözülmesini sağlar. Çok katmanlı, lineer, ileri beslemeli bir ağın yaptığı hesaplamalar, tek katmanlı lineer bir ağınkiyle aynıdır, çünkü lineer değişimin lineer değişimi, yine bir lineer değişimdir. Böylece, çok katmanlı lineer ileri beslemeli bir ağ, tek katmanlı lineer ileri beslemeli bir ağla aynı limitlere sahiptir. Kısaca, sadece giriş paternleri lineer bağımsızsa çalışır. Ancak, çok katmanlı lineer olmayan ileri beslemeli ağlarda bu sınırlama yoktur. Lineer olmayan aktivasyon fonksiyonları değişim gösterebildiği için, delta öğrenme kuralı kolayca bu tür çok katmanlı ileri beslemeli ağlara uyarlanabilir.

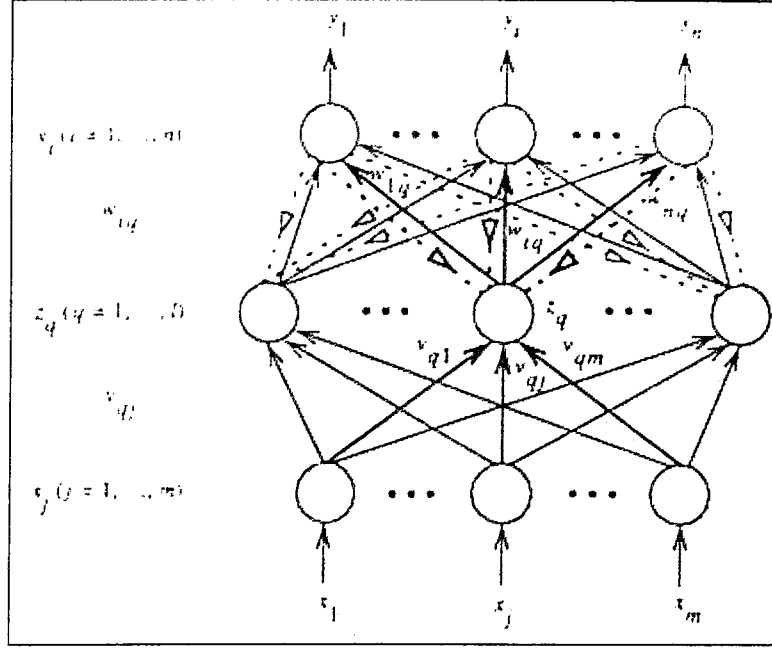
3.5 Çok Katmanlı İleri Beslemeli Ağlar

Tek katmanlı ağlarda, çözüm elde edilebilmesi için, giriş paternlerinin lineer ayrılabilir ya da lineer bağımsız olması gerekir. Basit perseptronlar için geçerli bu sınırlar, giriş ve çıkış katmanları arasında saklı katmanlar bulunan çok katmanlı ağlar için geçerli değildir. Üç katmanlı bir ağ, Şekil 3.7'de görülen XOR problemini çözmek için yeterli olabilmektedir. Çok katmanlı ağların problemleri çözmekteki büyük başarısı, önceden beri biliniyor olmasına rağmen, istenen bir fonksiyona nasıl uygulanacağı yakın zamanlarda anlaşılabilmiştir. Çok katmanlı ileri beslemeli ağların eğitimi için en önemli algoritma, geri yaymadır (back propagation).

3.5.1 Geri Yayma Algoritması

Geri yayma algoritması, yapay sinir ağlarının tarihsel gelişiminde en önemli noktalardan biridir. Bu algoritma, yapay sinir ağlarının bilimsel ve mühendislik alanlarında ve modelleme için kullanılmasında yeniden uyanışı sağlamıştır. Bu öğrenme algoritması, sürekli aktivasyon fonksiyonlarına sahip nöronlardan oluşan çok katmanlı, ileri beslemeli ağlara uygulanır. Geri yayma algoritmasının uygulandığı ağlara, geri yayma ağları adı da verilir. $\{(x^{(k)}, d^{(k)})\}$, $k = 1, 2, \dots, p$, gibi bir giriş-çıkış eğitme kümesi veildiğinde, bu algoritma geri yayma ağında, verilen giriş paternlerini doğru sınıflandıracak şekilde ağırlıkların değerlerini değiştirecek bir yöntem sunar. Bu ağırlık güncelleme algoritmasının temeli, basit perseptronlara uygulanan, eğim düşümü yöntemidir.

Verilen $(x^{(k)}, d^{(k)})$ giriş-çıkış çifti için, geri yayma algoritması iki fazlı bir veri akışı sağlar. Önce $x^{(k)}$ giriş paterni, giriş katmanından, çıkış katmanına doğru yayılır ve bu ileri veri akışının sonucu olarak, gerçekleşen çıkış değeri $y^{(k)}$ üretilir. Daha sonra $d^{(k)}$ ve $y^{(k)}$ arasındaki fark sayesinde üretilen hata sinyali, çıkış katmanından geriye doğru yayılarak, ağırlık değerleri güncellenir. Bu yöntem katman sayısı ne olursa olsun, her ağa uygulanabilir. Şekil (3.10)'da geri besleme algoritmasının kolaylıkla uygulanabileceği, üç katmanlı bir ağ görülmektedir. Burada, giriş katmanında m adet, saklı katmanda l adet, çıkış katmanında da n adet nöron bulunmaktadır. Sürekli çizgiler, sinyallerin ileri yayılmasını, kesikli çizgiler de hataların geri yayılmasını göstermektedir.



Şekil 3.10 Üç katmanlı, geri yaymalı ağ örneği

Ağa x giriş paterni uygulandığında, gizli katmandaki q nöronuna net giriş,

$$net_q = \sum_{j=1}^m v_{qj} x_j \quad (3.31)$$

olur ve

$$z_q = a(net_q) = a\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (3.32)$$

çıkışını verir. Böylece, çıkış katmanındaki i nöronuna net giriş,

$$net_i = \sum_{q=1}^l w_{iq} z_q = \sum_{q=1}^l w_{iq} a\left(\sum_{j=1}^m v_{qj} x_j\right) \quad (3.33)$$

olur ve

$$y = a(net_i) = a\left(\sum_{q=1}^l w_{iq} z_q\right) = a\left(\sum_{q=1}^l w_{iq} a\left(\sum_{j=1}^m v_{qj} x_j\right)\right) \quad (3.34)$$

çıkışını verir. Bu formüller, giriş sinyallerinin katmanlar boyunca ileri yayılmasını göstermektedir. Daha sonra, hata sinyalleri geriye doğru yayılır. Bunun için, formül (3.23)'teki gibi bir değer fonksiyonuna ihtiyaç vardır.

$$E(w) = \frac{1}{2} \sum_{i=1}^n (d_i - y_i)^2 = \frac{1}{2} \sum_{i=1}^n [d_i - a(net_i)]^2 = \frac{1}{2} \sum_{i=1}^n \left(d_i - a\left(\sum_{q=1}^l w_{iq} z_q\right) \right)^2 \quad (3.35)$$

Daha sonra, eğim düşümü yöntemine göre, saklı katmandan-çıkışa doğru olan ağırlıklar,

$$\Delta w_{iq} = -\eta \frac{\partial E}{\partial w_{iq}} \quad (3.36)$$

uygulanarak güncellenir. Daha sonra (3.33) - (3.35) formüllerini kullanarak,

$$\Delta w_{iq} = -\eta \left[\frac{\partial E}{\partial y_i} \right] \left[\frac{\partial y_i}{\partial net_i} \right] \left[\frac{\partial net_i}{\partial w_{iq}} \right] = \eta [d_i - y_i] [a'(net_i)] [z_q] = \eta \delta_{oi} z_q \quad (3.37)$$

elde edilir. Burada δ_{oi} hata sinyalidir ve çıkış katmanındaki i . nöronü gösterir. Hata sinyali,

$$\delta_{oi} = -\frac{\partial E}{\partial net_i} = -\left[\frac{\partial E}{\partial y_i} \right] \left[\frac{\partial y_i}{\partial net_i} \right] = [d_i - y_i] [a'(net_i)] \quad (3.38)$$

değerini alır. Burada net_i , çıkış katmanındaki i . nörona olan net giriş ve $a'(net_i)$ ise $\partial a(net_i) / \partial (net_i)$ 'dir. Buraya kadarki sonuçlar, formül (3.29)'da elde edilen tek katmanlı ağlar için olan delta kuralına çok benzemektedir. Ancak burada, perseptronun girişi, saklı katmanın çıkışı olan z_q 'dur.

Saklı katmandan - giriş katmanına doğru olan ağırlık güncellemeleri için de eğim düşümü metodu kullanılır ve gizli katmandaki q nöronunu, giriş katmanındaki j nöronuna bağlayan ağırlıktaki değişim miktarı bulunur.

$$\Delta v_{qj} = -\eta \left[\frac{\partial E}{\partial v_{qj}} \right] = -\eta \left[\frac{\partial E}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right] = -\eta \left[\frac{\partial E}{\partial z_q} \right] \left[\frac{\partial z_q}{\partial net_q} \right] \left[\frac{\partial net_q}{\partial v_{qj}} \right] \quad (3.39)$$

Formül (3.35)'ten rahatlıkla görülebilir ki, her hata terimi $[d_i - y_i]$, $i = 1, 2, \dots, n$, z_q 'nun bir fonksiyonudur. Böylece,

$$\Delta v_{qj} = \eta \sum_{i=1}^n [(d_i - y_i) a'(net_i) w_{iq}] a'(net_q) x_j \quad (3.40)$$

elde edilir. Formül (3.38)'i kullanarak, formül (3.40) şu şekilde yeniden yazılabilir:

$$\Delta v_{qj} = \eta \sum_{i=1}^n [\delta_{oi} w_{iq}] a'(net_q) x_j = \eta \delta_{hq} x_j \quad (3.41)$$

Burada δ_{hq} , saklı katmandaki q . Nöronun hata sinyalidir ve

$$\delta_{hq} = -\frac{\partial E}{\partial net_q} = -\left[\frac{\partial E}{\partial z_q} \right] \left[\frac{\partial z_q}{\partial net_q} \right] = a'(net_q) \sum_{i=1}^n \delta_{oi} w_{iq} \quad (3.42)$$

olarak tanımlanabilir. Burada net_q , saklı katmandaki q nöronuna net giriştir [Formül (3.31)]. Formül (3.38) ve formül (3.42)'de görüldüğü gibi, saklı katmandaki bir nöronun hata sinyali, çıkış katmanındaki bir nöronun hata sinyalinden farklıdır. Bu farkın sebebi, yukarıdaki ağırlık güncelleme yönteminin, genelleştirilmiş delta öğrenme kuralı olarak tanımlanmasıdır. Formül (3.42)'de görüldüğü gibi, saklı katmandaki nöron q 'nun hata sinyali δ_{hq} , çıkış katmanındaki nöronların hata sinyali δ_{oi} tarafından ve eğer besliyorsa, y_i tarafından yaratılır. Katsayılar, sadece ileri beslemede kullanılan ağırlıklardır ancak burada sinyalleri ileri yayacaklarına, hata sinyallerini (δ_{oi}) geri yaymaktadırlar. Bunlar, şekil (3.10)'daki kesikli çizgiler olarak görülebilir. Bu ayrıca, geriye yayma algoritmasının önemli bir özelliğini daha göstermektedir: Güncelleme kuralı yereldir; bu yüzden verilen bağlantıda ağırlık değişimini hesaplamak için, bu bağlantının her iki tarafındaki değerlere ihtiyaç vardır.

Görüldüğü gibi formül (3.37) ve formül (3.41), formül (3.13)'te görülen genel ağırlık öğrenme kuralının birer türüdür, sadece öğrenme sinyali, $r = \delta$ farklıdır. Bu formül, bir katmandan daha fazla saklı katmana sahip ağlar için de rahatlıkla türetilir. Özetle, belli bir sayıda katman için, geri besleme güncelleme kuralı,

$$\Delta w_{ij} = \eta \delta_i x_j = \eta \delta_{\text{çıkış-}i} x_{\text{giriş-}j} \quad (3.43)$$

şeklindedir. Burada çıkış- i ve giriş- j , i ve j nöronunun iki ucunu, x_j giriş ya da gizli nöronun giriş aktivasyonunu ve δ_i de formül (3.38)'de görülen çıkış katmanı için ya da formül (3.42)'de görülen diğer katmanlar için öğrenme ya da hata sinyalini göstermektedir. Aktivasyon fonksiyonu olarak, bipolar sigmoid fonksiyonu kullanıldığında,

$$\delta_{oi} = \frac{1}{2} (1 - y_i^2) [d_i - y_i] \quad (3.44)$$

ve

$$\delta_{hq} = \frac{1}{2} (1 - z_q^2) \sum_{i=1}^n \delta_{oi} w_{iq} \quad (3.45)$$

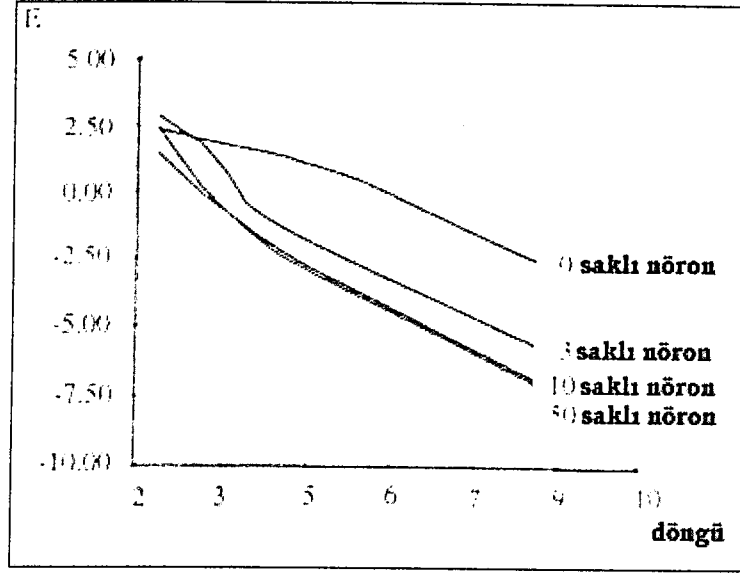
olur.

Geri yayma algoritmasının kullanıldığı ağlarda, başarının istenen düzeyde olmamasının sebebi, yetersiz eğitime, az sayıda saklı nöron ya da giriş ve istenen çıkış arasındaki deterministik ilişkinin eksikliği olabilir. Bu tür sorunların üstesinden gelebilmek için, bazen

dört, beş ya da daha fazla katmana ihtiyaç duyulabilir. Bunun nedeni, üç katmanlı bir yapıda çözüme ulaşabilmek için çok fazla sayıda gizli nörona ihtiyaç duyulurken, katman sayısının artırılmasıyla daha uygun sayıda nöron kullanılarak, istenilen çözüme ulaşılabilmesidir.

Geri yayma algoritması, ayırıcı olarak kullanıldığında, uygun bir eğitime sonrası her zaman uygun ağırlık değerlerini vermeyebilir. Ağırlık uzayındaki hata düzleminde, eğim düşümü sürer ve en yakın minimum hataya ulaştığında durur. Ancak, bu sadece giriş ve çıkış eğitime paternleri arasındaki ilişki deterministik olduğunda ve hata yüzeyi deterministik olduğunda doğrudur. Ne yazık ki gerçek hayattta, buna çok az rastlanır. Elde edilen karesel hata düzlemleri çoğunlukla rastlantısaldır. Bu, geri yayma algoritmasının, rastlantısal doğasıdır ve kaynağı, eğim düşümü yönteminin rastlantısallığıdır. Geri yayma algoritmasının, stokastik yaklaşımın özel bir türü olduğu kanıtlanmıştır. Diğer bir sorun ise, algoritmanın yerel minimumlara takılıp, tatmin edici sonuçlara ulaşamamasıdır. Bu geri yayma hata düzlemlerinin incelenmesini gerektirmektedir.

Hata fonksiyonları genellikle çok sayıda yerel minimum noktası içerir çünkü ağırlıkların alabildikleri değerlerin permutasyonu, ağırlık giriş-çıkış fonksiyonunu etkilemez. Bu, hata yüzeyinin çok sayıda çukura sahip olmasını sağlar. Ayrıca, yüzeylerin gerçek minimum değerlerinin çok yakınlarında yerel minimumlar vardır. Bu geri yayma algoritmasının, yerel minimumlara takılmasını sağlar. Ancak yapılan çalışmalarda görülmüştür ki, ihtiyaçtan daha fazla saklı birim kullanıldığında bu soruna rastlanmaz. Bunun sebebi, algoritmanın stokastik doğasıdır. Rastlantısallık, yerel minimumlardan kurtulmayı sağlar. Yapay sinir ağının boyutu büyüdükçe, stokastik faktörlerin sayısı da artar. Diğer bir sorun da hata yüzeylerindeki çok sayıdaki alan, birden fazla yönde eğime sahiptir. Bu genelde, bazı nöronların çıkış değerleri büyük olduğunda ve ağırlıklardaki küçük değişimlere fazla duyarlı olduğunda, sigmoid fonksiyonunun kuyruk kısımlarında görülür. Bu, yakınsamanın çok daha yavaş işlemlerini sağlar. Şekil 3.11'de, farklı sayıdaki saklı nöronla, geri yayma algoritmasındaki yakınsama görülmektedir. Kullanılan ağ, 23 girişe sahiptir ve tek saklı katmandaki saklı nöron sayısı değişmektedir. Eğitime kümesi, rastgele ve uniform seçilmiş 200 paterne sahiptir.



Şekil 3.11 Farklı sayıda saklı birimle, geri yayma algoritmasındaki yakınsama

Özetle geri yayma algoritması şu şekilde işler:

Geri Yayma Algoritması:

Q adet ileri beslemeli katmana sahip bir ağıımız olsun, $q = 1, 2, \dots, Q$ ve ${}^q net_i$ ve ${}^q y_i$ de sırasıyla q . katmandaki i . nöronun net girişi ve çıkışı olsun. Bu ağ, m adet giriş ve n adet çıkış nöronuna sahiptir. ${}^q w_{ij}$ de ${}^{q-1} y_j$ 'yi, ${}^q y_i$ 'ye bağlayan ağırlığı gösterebilir.

Giriş: Eğitim çifti kümesi: $\{(x^{(k)}, d^{(k)}) \mid k = 1, 2, \dots, p\}$. Bias işlemi için, giriş vektörünün boyutu bir artırılır ve bunun değeri -1 'e eşitlenir ($x_{m+1}^k = -1$).

Adım 0 (Başlatma) : $\eta > 0$ ve E_{max} (maksimum izin verilen hata) seçilir. Ağırlık değerleri rastgele olarak başlatılır. $k = 1$; $E^{(k)} = 0$ ve $E^{(k-1)} = 0$ olarak ayarlanır.

Adım 1 (Eğitme döngüsü) : k . giriş paterni giriş katmanına ($q = 1$) uygulanır:

$${}^q y_i = {}^1 y_i = x_i^{(k)}, \text{ bütün } i\text{'ler için.} \quad (3.46)$$

Adım 2 (İleri besleme) : ${}^q y_i$, çıkış katmanının bütün çıkış değerleri elde edilinceye kadar, her i ve q için eğitim sinyali ağda ileri beslenir:

$${}^q y_i = a({}^q net_i) = a\left(\sum_j {}^q w_{ij} {}^{q-1} y_j\right) \quad (3.47)$$

Adım 3 (Çıkış hatasının hesabı): Çıkış katmanı için hata değeri ve ${}^Q \delta_i$ hata sinyalleri hesaplanır:

$$E^{(k)} = \frac{1}{2} \sum_{i=1}^n (d_i^{(k)} - {}^q y_i)^2 + E^{(k-1)} \quad (3.48)$$

$${}^q \delta_i = (d_i^{(k)} - {}^q y_i) a'({}^q net_i) \quad (3.49)$$

Adım 4 (Hatayı geriye yayma): Ağırlıkları güncellemek ve önceki katmanlardaki hata sinyalleri $^{q-1}\delta_i$ 'leri hesaplamak için hata geriye yayılır:

$$\Delta^q w_{ij} = \eta^q \delta_i^{q-1} y_j \text{ ve } {}^q w_{ij}^{\text{yeni}} = {}^q w_{ij}^{\text{eski}} + \Delta^q w_{ij} \quad (3.50)$$

$${}^{q-1}\delta_i = a' \left({}^{q-1} \text{net}_i \right) \sum_j {}^q w_{ji} {}^q \delta_j, \quad q = Q, Q-1, \dots, 2 \text{ için} \quad (3.51)$$

Adım 5 (Bir döngü): Bütün eğitime setinin ağa beslenip beslenmediği kontrol edilir. Eğer $k < p$ ise, $k = k+1$ ve 1. adıma geri dönülür; aksi halde Adım 6'ya geçilir.

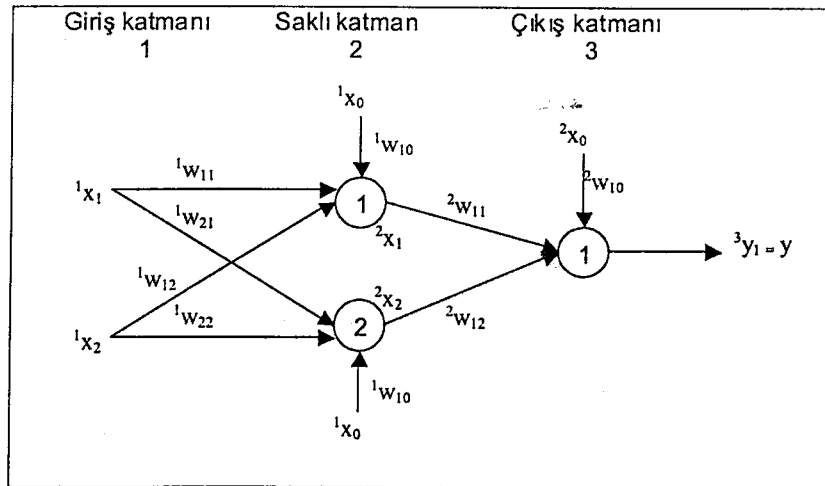
Adım 6 (Toplam hata kontrolü): Toplam hatanın kabul edilebilir olup olmadığı kontrol edilir. $E^{(p)} < E_{\max}$ ise, eğitime işlemi durdurulur ve son ağırlık değerleri gösterilir; aksi halde $k=1$, $E^{(k)} = 0$, $E^{(k-1)} = 0$ alınır ve Adım 1'e geri dönülerek yeni eğitime döngüsüne başlanır.

Geri Yaymayı Sonlandır

Geri yayma algoritmasının, Şekil (3.7)'de görülen XOR problemine uygulanması şöyledir:

Örnek 3.1 Geri yayma algoritmasının XOR problemine uygulanması

Ağ yapısı olarak, Şekil 3.12 'de görülen, iki giriş, iki saklı nöron ve tek çıkış nöronuna sahip ileri beslemeli bir ağ kullanılmıştır. Geri yayma algoritması, sonuca ulaşmak için binlerce tekrarlı eğitime döngüsüne ihtiyaç duyduğu için, burada sadece bir eğitime adımı gösterilmiştir. Her katmandaki giriş değerleri, ${}^q x_i$ olarak gösterilmiştir. Burada q , katmanı, i ise nöronu simgeler. Örneğin, ağa girişler ${}^1 x_1$ ve ${}^1 x_2$ ile ve gizli katmanın çıkışları da ${}^2 x_1$ ve ${}^2 x_2$ ile gösterilmiştir. Çıkış nöronunun çıkışı da ${}^3 y_n$ (ya da y) ile gösterilmiştir. Bias değerleri de, saklı katman için ${}^1 x_0$ ve çıkış katmanı için ${}^2 x_0$ ile ifade edilir.



Şekil 3.12 XOR probleminin çözümünde kullanılan ağ yapısı

Ağırlık değerleri w_{ij} ile gösterilmiştir. Burada j , giriş vektörünü, i de bağlantı yapılan nöronu simgelemektedir. $j = 0$ olduğunda, bu da bias değerini göstermektedir.

Öncelikle, ağırlık değerleri -1 ile 1 arasında rastgele değerlerden başlatılmıştır:

$$1. \text{ saklı nöron için: } {}^1w_{10} = 0.341232, {}^1w_{11} = 0.129952, {}^1w_{12} = -0.923123$$

$$2. \text{ saklı nöron için: } {}^1w_{20} = -0.115223, {}^1w_{21} = 0.570345, {}^1w_{22} = -0.328932$$

$$\text{Çıkış nöronu için: } {}^2w_{10} = -0.993423, {}^2w_{11} = 0.164732, {}^2w_{12} = 0.752621$$

İlk eğitime verisi olarak, ${}^1x_0 = 1$ (bias), ${}^1x_1 = 0$, ${}^1x_2 = 0$ seçilmiştir. Böylece saklı katmandaki nöronların net girişleri şu değerleri almıştır:

$$\text{Nöron 1 : } (1 * 0.341232) + (0 * 0.129952) + (0 * -0.923123) = 0.341232$$

$$\text{Nöron 2 : } (1 * -0.115223) + (0 * 0.570345) + (0 * -0.328932) = -0.115223$$

Daha sonra, saklı katmandaki nöronların çıkış değerleri hard limit (eşik) fonksiyonundan geçirilmiştir:

$${}^2x_1 = 1 / (1 + e^{(-0.341232)}) = 0.584490$$

$${}^2x_2 = 1 / (1 + e^{(0.115223)}) = 0.471226$$

Saklı katman için çıkış değerleri elde edilmiştir. Bu değerler, aynı zamanda çıkış katmanı için giriş değerleridir (${}^2x_0 = 1$ (bias), ${}^2x_1 = 0.584490$, ${}^2x_2 = 0.471226$). Aynı hesaplamalar, çıkış katmanı için yapıldığında:

$$\text{Çıkış nöronunun net girişi : } (1 * -0.993423) + (0.584490 * 0.164732) + (0.471226 * 0.752621) = -0.542484$$

$${}^3y_3 = y = 1 / (1 + e^{(-0.542484)}) = 0.367610$$

Bu değer, ağın elde edilen çıkış değeridir. Oysa ki XOR probleminde 0 ve 0 giriş değerleri için istenen çıkış değeri 0'dır. Bu hatanın geriye yayılarak, düzeltilmesi gerekmektedir. Bunun için de delta değerlerinin hesaplanması gerekir. Çıkış katmanı için hata değeri ve hata sinyali sırasıyla :

$$E^{(1)} = \frac{1}{2}(0 - 0.367610)^2 + 0 = 0.067569$$

$$\begin{aligned}
{}^3\delta_1 &= y(1-y)(d-y) \\
&= 0.367610 * (1 - 0.367610) (0 - 0.367610) \\
&= - 0.085459
\end{aligned}$$

olarak elde edilir. Bu değer, hatanın geriye yayılması için kullanılabilir :

$$\begin{aligned}
{}^2\delta_1 &= {}^2x_1 (1 - {}^2x_1) {}^2w_{11} {}^3\delta_1 \\
&= 0.584490 * (1 - 0.584490) * (0.164732) * (-0.085459) = - 0.0034190 \\
{}^2\delta_2 &= {}^2x_2 (1 - {}^2x_2) {}^2w_{12} {}^3\delta_1 \\
&= 0.471226 * (1 - 0.471226) * (0.752621) * (-0.085459) = - 0.0160263
\end{aligned}$$

Bu delta değerleri, ağırlıkları güncelleme için kullanılır. Bu örnekte, η değeri olarak 0.5 seçilmiştir. Bazı ağırlık değişimi değerleri 0 (sıfır) olacaktır, çünkü giriş değerleri 0 seçilmiştir. Böylece, ağırlık değişimleri :

$$\begin{aligned}
\Delta^1 w_{10} &= \eta * {}^1x_0 * {}^2\delta_1 = 0.5 * 1 * - 0.0034190 = - 0.017095 \\
\Delta^1 w_{11} &= 0 \\
\Delta^1 w_{12} &= 0 \\
\Delta^1 w_{20} &= 0.5 * 1 * - 0.0160263 = - 0.0080132 \\
\Delta^1 w_{21} &= 0 \\
\Delta^1 w_{22} &= 0 \\
\Delta^2 w_{10} &= 0.5 * 1 * - 0.0085459 = - 0.042730 \\
\Delta^2 w_{11} &= 0.5 * 0.584490 * - 0.0085459 = - 0.024975 \\
\Delta^2 w_{12} &= 0.5 * 0.471226 * - 0.0085459 = - 0.020135
\end{aligned}$$

değerlerini alır. Bu değerler, ilk ağırlık değerlerine eklenerek, eğitim sonundaki ağırlık değeri bulunur:

$$\begin{aligned}
{}^1w_{10} &= 0.341232 - 0.017095 = 0.294137 \\
{}^1w_{11} &= 0.129952 + 0 = 0.129952 \\
{}^1w_{12} &= - 0.923123 + 0 = - 0.923123 \\
{}^1w_{20} &= - 0.115223 - 0.0080132 = - 0.1232362 \\
{}^1w_{21} &= 0.570345 + 0 = 0.570345
\end{aligned}$$

$${}^1w_{22} = -0.328932 + 0 = -0.328932$$

$${}^2w_{10} = -0.993423 - 0.042730 = -1.036153$$

$${}^2w_{11} = 0.164732 - 0.024975 = 0.139757$$

$${}^2w_{12} = 0.752621 - 0.020135 = 0.732486$$

Bu örnek, www.generation5.org/XORnet.html adresinden alınmıştır.

3.5.2 Geri Yayma Algoritmasının Öğrenme Faktörleri

Geri yayma algoritmasında yakınsama, başlangıç ağırlıkları, öğrenme sabiti, değer fonksiyonu, güncelleme kuralı, eğitime kümesinin boyutu ve doğası ile ağıın yapısı (katman sayısı ve katmanlardaki nöron sayısı) gibi önemli öğrenme faktörlerine bağlıdır. Bu faktörler, öğrenme hızı, yerel minimum ve geri yayma ağlarının genelleme kapasitesi üzerinde etkilidir.

3.5.2.1 Başlangıç Ağırlıkları

Çok katmanlı, ileri beslemeli bir ağda, başlangıç ağırlıkları, çözümü büyük ölçüde etkilemektedir. Genellikle, küçük ve rastlantısal değerlerden başlatılırlar. Eğer çözüm farklı ağırlık değerleri gerektiriyorsa, başlangıç ağırlıklarının eşit olması, ağın eğitiminin hatalı olmasına yol açar. Başlangıç ağırlıkları büyük olamaz, aksi durumda sistem yerel minimumlara ya da başlangıç noktasına çok yakın düz bir düzleme takılır. Ağırlık w_{ij} 'yi seçmenin uygun bir yolu, $[-3/\sqrt{k_i}, 3/\sqrt{k_i}]$ aralığından seçmektir. Burada k_i , i nöronuna olan giriş adedidir.

3.5.2.2 Öğrenme Sabiti

Geri yayma algoritmasının etkinliğini ve yakınsamasını etkileyen diğer önemli bir faktör, öğrenme sabiti η 'dir. Farklı öğrenme durumları için tek bir uygun değer yoktur ve η genelde her problem için deneme yanılma yöntemiyle seçilir. Büyük η değeri, yakınsamayı hızlandırır fakat, hedefi ıskalama yol açabilir. 10^{-3} ila 10 arasındaki değerler çoğu geri yayma problemi için uygundur.

Öğrenme sabitindeki diğer bir sorun da, başlangıç için ideal olan değerlerin, öğrenmenin ilerki aşamalarında uygun olmadığıdır. Bu yüzden, daha etkili bir yaklaşım şekli, adaptif öğrenme sabiti kullanmaktır. Bu yöntem, belirli bir ağırlık güncellemesinin, değer fonksiyonunda bir düşüşe sebep olup olmadığını araştırılmasına dayanır. Eğer düşüşe sebep olmuyorsa, hedef ıskalanmış demektir ve η değerinin küçültülmesi gerekmektedir. Diğer taraftan, eğer üstüste birkaç adım değer fonksiyonu azalıyor, η arttırılmalıdır. Başka bir deyişle, öğrenme sabiti şu kurala göre güncellenmelidir:

$$\Delta\eta = \begin{cases} +a & \Delta E < 0 \text{ sürekli} \\ -b\eta & \Delta E > 0 \\ 0 & \text{aksi halde} \end{cases} \quad (3.52)$$

Burada, ΔE , değer fonksiyonundaki değişimdir ve a ile b de pozitif katsayılarıdır.

3.5.2.3 Değer Fonksiyonu

Formül (3.35)'teki quadratik değer fonksiyonu tek uygun seçenek değildir. Karesel hata terimi, $(d_i - y_i)^2$, şartlar aynı olduğunda minimize olan diğer bir $F(d_i, y_i)$ diferansiyel fonksiyonuyla değiştirilebilir. Bu yeni değer fonksiyonu baz alınarak, güncelleme kuralı yaratılabilir. Kolaylıkla görülebilir ki, farklı değer fonksiyonları için, sadece formül (3.38)'de görülen çıkış katmanındaki hata sinyali, δ_{oi} değişmektedir. Geri yayma algoritmasının diğer bütün eşitlikleri aynı kalır.

Genellikle kullanılan değer fonksiyonları, matematiksel formülasyonun kolaylığı için, L_p normunda ($1 \leq p \leq \infty$) olanlardır. Bu değer fonksiyonları şu şekildedir:

$$E = \frac{1}{p} \sum_i (d_i - y_i)^p, \quad 1 \leq p \leq \infty \quad (3.53)$$

Genelde, quadratik değer fonksiyonunda kullanılan en küçük kareler kriteri (L_2 normu), basitliği yüzünden tercih edilir. Diğer bir norm da Chebyshev normu olarak bilinen L_∞ normudur. L_∞ normu, paternler için hatayı,

$$E^\infty = \sup_i |d_i - y_i| \quad (3.54)$$

olarak tanımlar. Burada, $\sup|\cdot|$, vektörün en büyük terimini seçen fonksiyonu göstermektedir. Bu tanım, bütün hata ölçütü E^∞ 'un, hata vektöründeki en büyük elemana eşit olduğunu ,

diğer bütün elemanların ihmal edilebilir olduğunu varsayar. Formül (3.54)'teki değer fonksiyonundan, çıkış katmanı için hata sinyali δ_{oi} bulunabilir:

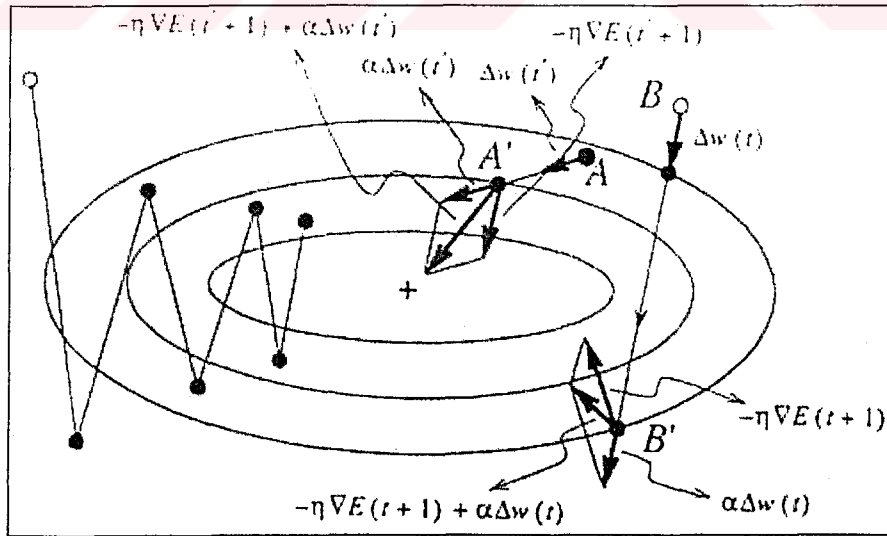
$$\delta_{oi} = -\frac{\partial E^{\infty}}{\partial net_i} = \begin{cases} 0, & i \neq i^* \\ a'(net_i^*) \text{sgn}(d_i^* - y_i^*), & i = i^* \end{cases} \quad (3.55)$$

Burada, i^* , çıkış hata vektörünün en büyük elemanının değeridir. Hata sinyali, çıkış birimleri arasında sadece en büyük olanının değeridir ve i . nörondan geriye doğru yayılır. Diğer çıkış birimlerinden hata geriye yayımı olmaz.

3.5.2.4 Momentum

Öğrenme sabiti η , çok küçükse, eğim düşümü çok yavaş olur ve çok büyükse osilasyona sebep olabilir. Etkili ve sık kullanılan bir yöntem, normal eğim düşümü yöntemine momentum teriminin eklenmesiyle, osilasyona meydan vermeden, büyük öğrenme sabitlerinin kullanılmasıdır. Bu yöntemde, her ağırlık değişimine, bir önceki değişimden bir pay eklenerek yapılır.

$$\Delta w(t) = -\eta \nabla E(t) + \alpha \Delta w(t-1) \quad (3.56)$$



Şekil 3.13 Basit bir düzlemde eğim düşümleri. Solda momentum terimi yoktur, sağdakilerde vardır.

Burada $\alpha \in [0,1]$ momentum parametresidir ve genellikle 0,9 değeri kullanılır. Şekil 3.13'te basit bir düzlemde momentumlu ve momentumlu eğim düşümleri görülmektedir. Momentumlu işlemlerin (soldaki), momentumlulardan (sağdakiler) daha fazla osilasyon

yaptığı görülebilir. Eğer ağırlık değişimi doğru yönde ise (Şekil 3.13'te A dan A' ne doğru), momentumun işlemi hızlandığı görülebilir. Diğer taraftan, ıskalama durumunda, hareketi hedef noktaya doğru yönlendirebilir (Şekil 3.13'te B den B' ne doğru). Bu gözlem, momentum teriminin yakınsamayı hızlandığı ve yeterli ve güvenli bir öğrenme profili sunduğunu gösterir.

3.5.2.5 Güncelleme Kuralları

Eğim düşümü metodu en basit optimizasyon tekniklerinden biri olmakla birlikte, çok etkili bir yöntem değildir. Geri yayma algoritmasının gözle görülebilir biçimde daha hızlı yakınsaması için, numerik optimizasyon teorisi uygulanabilir. Numerik optimizasyon teorisi, yapay sinir ağlarında öğrenme oranını arttıracak zengin ve sağlam teknikler sunar.

Eğim düşümü metodu, hata fonksiyonunun sadece birinci kuvvetini azaltmayı düşünür. Daha büyük mertebeden türevleri göz önüne almak faydalı olabilir. Mevcut nokta w_0 'da, $E(w)$ üzerinde Taylor serisi uygulanırsa:

$$E(w) = E(w_0) + (w - w_0)^T \nabla E(w_0) + \frac{1}{2} (w - w_0)^T H(w) (w - w_0) + \dots, \quad (3.57)$$

Burada $H(w)$, Hess matrisi diye bilinir ve w_0 'a göre 2. dereceden türevi alınır:

$$H(w) = \nabla^2 E(w) \quad \text{veya} \quad H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} \quad (3.58)$$

bulunur. $E(w)$ 'nin minimumunu bulmak için, gradyanı sıfıra eşitlenir:

$$E(w) = \nabla E(w_0) + H(w)(w - w_0) + \dots = 0 \quad (3.59)$$

Eğer üçüncü ve daha büyük seviyedeki terimler ihmal edilirse,

$$w = w_0 - H^{-1}(w) \nabla E(w_0) \quad (3.60)$$

elde edilir. k. eğitime adımını göstermek için, k indisi kullanılırsa,

$$w^{(k+1)} = w^{(k)} - H^{-1}(w^{(k)}) \nabla E(w^{(k)}) \quad (3.61)$$

elde edilir. Bu, Newton'un ağırlık güncelleme yöntemi olarak bilinir. Newton'un yöntemi, ikinci dereceden türevi, gradyana ekleyerek, bir sonraki adımın yönünü ve boyutunu belirler. Konveks fonksiyonlarda, sonuca yaklaştıkça quadratik olarak yakınsar. Ancak, Newton'un

yönteminde bazı sakıncalar vardır. Bunlardan ilki, yakınsama için, sonucun önceden iyi bir şekilde tahmin edilmesi gerekmektedir. İkincisi, konveks fonksiyonlar için, hızlıca yakınsar, ancak konveks olmayan fonksiyonlarda, kolaylıkla yerel minimum noktalarına takılabilir. Üçüncü ve en önemli sakınca, her iterasyon, Hess matrisinin ve tersinin hesaplanmasını gerektirir. Bu yüzden pratik bir teknik değildir ve alternatifleri ya da farklı varyasyonları önerilir.

3.5.2.6 Eğitim Verisi ve Genelleştirme

Eğitim verisinin her zaman yeterli ve düzenli olması gerekir. Ancak, eğitim verisinin seçimi için her zaman işe yarayan bir yöntem ya da uygun bir kural yoktur. Bir yöntem, eğitim kümesinin beklenen bütün giriş değerlerini kapsaması ve eğitim işlemi boyunca eğitim vektör çiftlerinin kümeden rastgele seçilmesidir. Daha açıkça, giriş düzleminin, M adet ayrık, lineer bağımsız bölgeden oluştuğu varsayalım. P de eğitim paterni sayısındaki alt sınır olsun. Böylece, P 'yi $P/M \gg 1$ olacak şekilde seçerek, farklı hiperdüzlem parçalarındaki paternlerden oluşan eğitim setiyle eğitilmesi sağlanır. Bazı durumlarda, eğitmeye yardımcı olmak için ölçeklendirme ya da normalizasyon gerekebilir. Örneğin, çıkış fonksiyonu sigmoid ise, çıkış değerleri uygun biçimde ölçeklenmelidir.

Geri yayma ağları genelleştirme için uygundur. Ağ, yeni girilen giriş paternlerine uygun cevaplar verdiğinde genelleştirme yapılmış demektir. Verilen sayıda eğitim verisi için, çok fazla eğitilebilir parametreye sahip ağlar, öğrenmeyi iyi yaparlar ama genelleştirmeyi iyi yapamazlar. Bu durum aşırı uydurma olarak bilinir. Çok az eğitilebilir parametreyle, ağ eğitmeyi tamamlayamaz ve test kümesinde kötü sonuç verir. Ağın eğitim veri kümesinden, test veri kümesine genelleştirmesini iyileştirebilmek için, çıkış elemanlarını değiştirmeyecek şekilde, giriş uzayındaki paternlerde ufak değişiklikler yapılması gerekir. Bu, eğitim kümesine, giriş uzayındaki paternlerin farklı varyasyonlarını ekleyerek yapılır ancak çok zahmetlidir.

3.5.2.7 Saklı Nöron Sayısı

Saklı katmanın boyutu, gerçek hayattaki problemleri, çok katmanlı, ileri beslemeli yapay sinir ağlarına uygulamakta karşılaşılan temel sorudur. Bu konunun kesin analizi çok zordur, çünkü

ağ eşlemi karmaşıktır ve bitirilmiş eğitime işlemlerinin doğası deterministik değildir. Bu yüzden, saklı katmanın boyutu genellikle denemeyle bulunur. Genel bir yöntem şöyledir: Makul boyutta bir ağ için (örneğin yüzlerce ya da binlerce giriş), saklı katmandaki nöron sayısı giriştekenden az bir miktar daha azdır. Eğer ağ, çözüme yakınsamayı sağlamıyorsa, daha fazla saklı nöron gerekebilir. Eğer yakınsama sağlanıyorsa, daha az sayıda saklı nöron kullanmak denenebilir ve istenilen sistem performansı elde edilebilir.

Analitik analizler temel alınarak yapılan bazı yaklaşımlar da vardır. “Farklı bölgelerden, farklı giriş değerlerine sahip olacak şekilde giriş uzayını parçalamaya yetecek kaç saklı nörona ihtiyaç vardır ?” sorusuna cevap aranır. m boyutlu, arttırılmamış giriş uzayının, aynı bölgedeki elemanlar aynı çıkışı verecek şekilde (örneğin aynı sınıfa ait), M adet ayrık ve her bölümü hiperuzayın bir parçası olan bölgeye ayrıldığı varsayılınsın. Tek saklı katmana sahip ağlar m boyutlu giriş uzayında keyfi karar bölgeleri yaratabilmelerine karşın, burada sorun, ağın istenilen düzeyde çalışması için gerekli saklı nöron sayısı, N_m 'yi bulmaktır. M , N_m ve m arasındaki ilişki şöyle gösterilmiştir:

$$N_m + 1 \leq M \leq \sum_{j=0}^m \binom{N_m}{j}, \quad N_m < j \text{ için, } \binom{N_m}{j} = 0 \quad (3.62)$$

N_m adet gizli nöron kullanarak, m boyutlu giriş uzayında bulunabilecek lineer ayrılabilen bölgelerin maksimum sayısı, $N_m > m$ için,

$$M_{\max} = \sum_{j=0}^m \binom{N_m}{j} = 1 + N_m + \frac{N_m(N_m-1)}{2!} + \dots + \frac{N_m(N_m-1)\dots(N_m-m+1)}{m!} \quad (3.63)$$

olarak bulunur. $N_m \leq m$ için,

$$M_{\max} = 2^{N_m} \quad \text{ve} \quad N_m = \log_2 M_{\max} \quad (3.64)$$

dir. Örnek olarak, Şekil 3.7’de görülen XOR problemi alındığında, iki boyutlu giriş patenleri için, $M = 3$ ayrık bölge vardır. Tek saklı katmanda iki nörona sahip bir ağ, bu problemi çözmek için yeterlidir.

(Bölüm 3 içinde verilen bilgiler ve şekiller, Lin ve Lee (1996) “*Neural Fuzzy Systems; A Neuro-Fuzzy Synergism to Intelligent Systems*” adlı kaynaktan alınmıştır.)

4. TRENLERARASI ÇATIŞMALARIN ÇÖZÜMÜ İÇİN YAPAY SİNİR AĞI MODELİ

4.1 Giriş

Günümüzde, trenlerarası çatışmaların çözümü dispeçerler tarafından yapılmaktadır. Dispeçerler, yıllar boyu kazandıkları bilgi ve deneyimleri sayesinde, çatışmaların çözümünü gerçekleştirmektedirler. Çatışmaların çözülmesi gerçekten zor ve deneyim gerektiren bir işlemdir; çünkü trenler çatışmaya girdiklerinde, önceden belirlenmiş olan programda çok büyük boyutta sapmalar olabilmektedir. Çatışma çözümlerinde doğru kararları verebilen kişileri bulmak oldukça zordur. Ayrıca, deneyimsiz dispeçerleri yetiştirmek de çok uzun zaman gerektiren bir iştir. Bu nedenlerle, bir dispeçerin davranışını daha önceki olaylara dayanarak taklit edecek otomatik bir sistem geliştirmek çok faydalı olacaktır. Bunun için geçmişteki çatışmaların çözümlerinin detaylı bir biçimde incelenmesi gerekmektedir.

Önceden hazırlanmış çizelgelerden sapma olması, trenlerarası çatışmaları doğurmakta bu da karar verme zorunluluğunu ortaya çıkarmaktadır. Dispeçerin görevi de bu çatışmaları, trenlerin özelliklerini/niteliklerini göz önüne alarak çözmektir. Ancak iki tren arasında gerçekleşen çatışmanın optimum çözümüne ulaşmak, pratik olarak imkansız olduğu için dispeçer bazı basitleştirici kurallar kullanarak bu çatışmaları çözmektedir. Örneğin, işletme tarafından trenlere atanmış temel öncelik sayıları, bir çok durumda iki alternatifte sahip çatışmaların çözüm sayısını bire indirmekte, böylece dispeçerin karar verme işlevi basitleştirilmiş olmaktadır. Bu çalışmada dispeçerin, trenlerarası çatışmaların çözümünde uyguladığı karar davranışının çok nitelikli/kriterli modelinin oluşturulması amaçlanmış, bu işlem için de yapay sinir ağı tekniği kullanılmıştır.

Modelin geçerliliğini test etmek amacıyla, Türkiye Cumhuriyeti Devlet Demiryolları (TCDD)'nden gerçek trafik hareketlerine ilişkin veriler elde edilmiştir. Test bölgesi olarak, İstanbul-Ankara demiryolu bağlantısının **tek hatlı** olan Arifiye-Çukurhisar (Eskişehir) kesimi seçilmiştir (İnönü – Çukurhisar arasındaki yaklaşık 14.3 km uzunluğundaki kesim çift hatlıdır.) Uzunluğu yaklaşık 163 kilometre olan ve hat boyunca 17 ara istasyon/sayding bulunan bu kesimin seçilmesinin önemli bir nedeni vardır. İstanbul ve Ankara terminallerinden kalkan trenler 00:00 - 06:00 zaman aralığında bu bölge içinde buluşmaktadırlar. Bu durum, tren dispeçerleri için, trenlerarası çatışmaların çözülmesine yönelik bir çok kararın alındığı,

oldukça yoğun bir mesai anlamına gelmektedir. Bu çalışmada, 21.12.2001 – 31.12.2001 ve 11.07.2002 – 20.07.2002 tarihleri arasındaki 21 farklı işletme gününe ait tren-graflardan faydalanılmıştır (TCDD, 1995 ve TCDD, 2000). Çalışmada sadece **karşılaşma çatışmaları** dikkate alınmıştır. Seçilen zaman aralığında 331 adet bu tür çözülmüş trenlerarası çatışma modelde tanımlanmıştır. Malavasi ve Ricci (2001), örnek sayısı (E), giriş parametreleri (F) ve çıkış parametresinin alabileceği değerler (N) arasındaki ilişkiyi gösteren R katsayısının ($R = E / (F + N)$) en az 10 olması gerektiğini belirtmektedir. Bu çalışma kapsamında, R değeri, $331 / (8 + 2) = 33.1$ değerini almaktadır. Buradan da anlaşılacağı gibi, örnek sayısı, problemin modellenmesi için yeterlidir. Ağ öncelikle, 173 örnek tarafından eğitilmiş, daha sonra da kalan 158 örnekle test edilmiştir.

4.2 Dispeçerin Karar Değişkenleri: Çatışan Trenlerin Nitelikleri

Tren dispeçerleri, trenlerarası çatışmaların çözümünde yıllar boyunca kazandıkları bilgi ve deneyimlerini kullanırlar. Çözüm kararlarını alırken trenlerin yer ve durumlarına ait farklı özelliklerden faydalanmaları gerekmektedir. Ancak yapılan çalışmalar göstermektedir ki, insan beyni karşılaştırma yaparken en fazla yedi kriteri gözönünde bulundurabilir. Bu nedenle, dispeçer kararlarını modellerken, dispeçerin en çok kullandığı karar değişkenlerinin neler olduğu incelenmiştir. Bu çalışma kapsamında, karar değişkeni (giriş parametresi) olarak **çatışan her iki tren** için, *temel öncelik sayısı*, *kritik oran*, *miyopik çözümdeki gecikme* ve *aktif çatışmadan sonraki potansiyel çatışma sayısı* alınmıştır. Çalışma kapsamında, ihtiyari trenlere ait yukarıda sayılan kriterlerden bazıları belirlenemediği için, bu trenlerin girdiği çatışmalar modele dahil edilmemiş, ancak potansiyel çatışma sayısında kullanılmıştır. Dispeçerin çözümü 1 ise, modelin çözümü 0.5'den büyük olduğunda, dispeçer ile modelin kararlarının uyumlu olduğu; dispeçerin çözümü 0 ise, modelin çözümü 0.5'den küçük olduğunda, dispeçer ile modelin kararlarının uyumlu olduğu söylenebilir.

4.2.1 Temel Öncelik Sayısı

Temel öncelik sayısı, her trenin sınıfına/hızına bağlı olarak atanan sabit bir sayıdır. Genel kural, en hızlı trene en küçük sayının atanması şeklindedir. Bu çalışma kapsamında ele alınan trenler için temel öncelik katsayıları Çizelge 4.1'de verilmiştir.

Çizelge 4.1 Temel öncelik sayıları

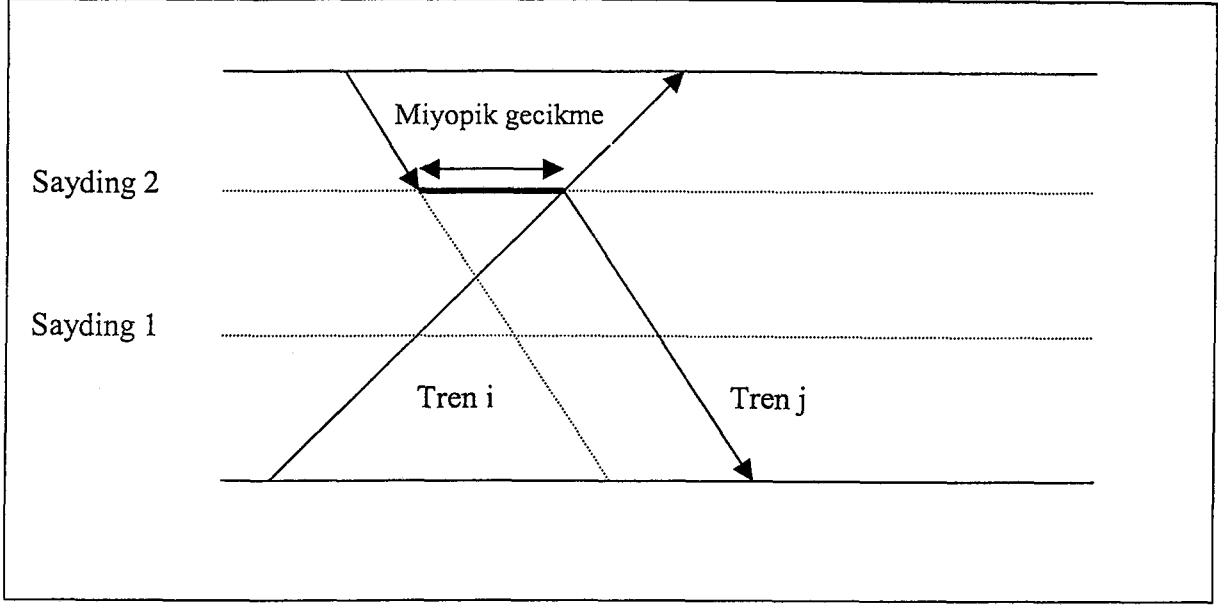
Tren Numarası	Tren Adı	Temel Öncelik Sayısı
11021-11022	Fatih Ekspresi	20
11125-11126	İç Anadolu Mavi Treni	20
11207-11208	Anadolu Ekspresi	50
11209-11210	Ankara Ekspresi	50
11319-11320	Meram Ekspresi	60
11321	Pamukkale Ekspresi	60
11512	Trans Asya Ekspresi	70
11532-11542	Van Gölü Ekspresi/Güney Ekspresi	70
12517	Yolcu Treni	80
14001	Hızlı Yük Treni	100
14005	Hızlı Yük Treni	100
14108	Hızlı Yük Treni	100

4.2.2 Kritik Oran

Kritik oran, trenin gecikmiş olup olmadığının bir göstergesidir. Bu oran, trenin planlanmış son varış zamanına kalan sürenin, kalan en küçük seyir süresine oranı olarak tanımlayan dinamik bir niteliktedir. Bir'den küçük bir kritik oran, bir acil durum göstergesi olarak, trenin çizelgeden daha fazla geri kalmasını önlemek için, kısıtlamasız ilerlemesi gerektiği anlamına gelir.

4.2.3 Miyopik Çözümdeki Gecikme

Miyopik çözümdeki gecikme, trenin girdiği çatışmanın kendi aleyhine çözülmesi durumunda, diğer trenin yolu boşaltıncaya kadar, bu trenin buluşma noktasında bekletilme süresidir.



Şekil 4.1 Tren i'nin miyopik çözümdeki gecikmesi

4.2.4 Potansiyel Çatışma Sayısı

Trenin aktif çatışmadan sonra, son varış noktasına kadar, girebileceği potansiyel çatışmaların sayısı, modele potansiyel çatışma sayısı olarak dahil edilmiştir. İhtiyari trenlerin girdiği çatışmalar, modelde kullanılmamasına rağmen, bu çatışmalar, potansiyel çatışma sayısına dahil edilmiştir.

4.3 Yapay Sinir Ağı Modelleri

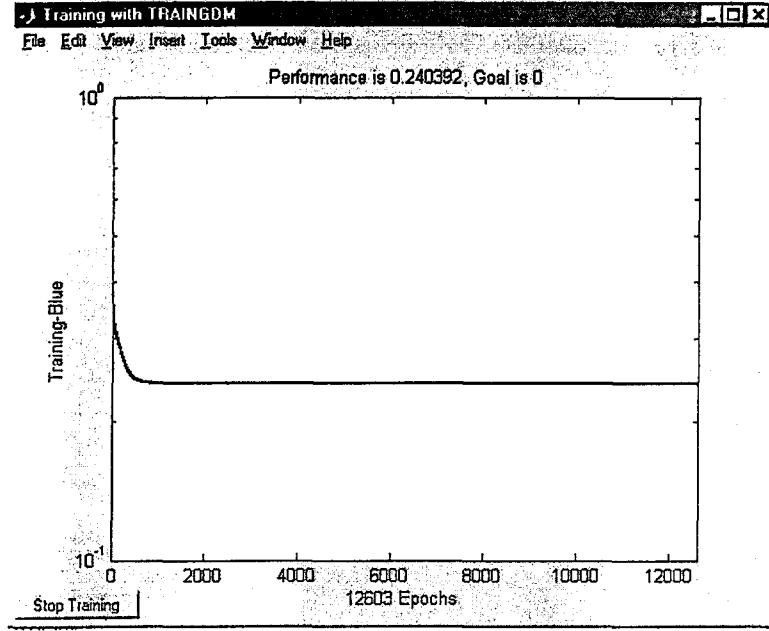
Kurulan yapay sinir ağı modellerinde, giriş değerleri olarak çatışan her iki tren için, temel öncelik sayısı, kritik oran, miyopik çözümdeki gecikme ve aktif çatışmadan sonraki potansiyel çatışma sayısı kullanılmıştır. Tren numarası küçük olan trenin değerleri ilk önce girilmiştir. Sinir ağının çıkış değeri olarak, küçük numaralı tren geçiyor, büyük numaralı tren bekletiliyorsa 1, aksi durumda 0 değerini vermesi istenmiştir. Çözümünün 1 ve 0 değerlerini alması istenen çatışmalar modele şaşırtmalı olarak girilmiştir. Hesaplamalarda kolaylık sağlaması için, giriş değerleri normalize edilerek ağa tanıtılmıştır. Eşik fonksiyonu olarak unipolar sigmoid fonksiyonu kullanılmıştır. Hesaplamalar, 256 MB bellekli, Pentium IV, 2 GHz bilgisayarda yapılmış ve Matlab 6.5 (Neural Networks Toolbox Version 4.0.2) programı kullanılmıştır.

4.3.1 Normalizasyon

Temel öncelik sayısı, merteye itibarıyla miyopik çözümdeki gecikme ve potansiyel çatışma sayısından, bunlar da kritik orandan büyüktür. Bu mertebelerdeki farklılık, ağırlıkların güncellenmesi sırasında sorun yaratabileceği ve çözüm için gerekli döngü sayısının artmasına yol açabileceği için, bu değerler 0 - 1 arasında değerler alacak şekilde normalize edilmiştir. Kullanılan kriterlerden “temel öncelik sayısı” ve “kritik oran”, çatışan trenin “geçen tren” olabilmesi için “ne kadar küçükse o kadar iyi” sınıfına girmekte; kriterlerden “miyopik gecikme” ve “potansiyel çatışma sayısı” ise “ne kadar büyükse o kadar iyi” sınıfına girmektedir. Buna göre, normalizasyon, “ne kadar küçükse o kadar iyi” kriteri için, aldığı değerlerin (sıfırdan büyük) en küçüğü belirlenir ve bu en küçük değer, her kriter değerine bölünür. “Ne kadar büyükse o kadar iyi” kriteri için, aldığı değerlerin en büyüğü belirlenir ve her kriter değeri bu en büyük değere bölünür. Bu sayede, giriş vektörlerinin her biri 0 – 1 arasında değerler almışlardır.

4.3.2 Tek Katmanlı İleri Beslemeli Model

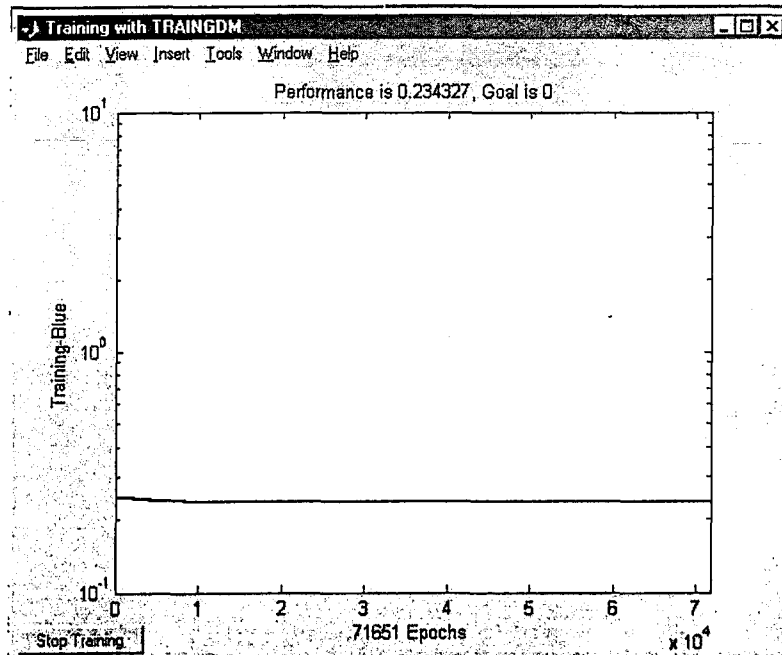
Problemin lineer bir çözüme sahip olup olmadığının incelenmesi amacıyla, önce tek katmanlı ileri beslemeli bir ağ yapısı kurulmuştur. 8 giriş vektörü/parametresi ve tek çıkış nöronuna sahip bu modelde, karesel hata (E), 0.240392'ye kadar düşürülebilmiştir (Şekil 4.2). Ancak bu değer, modelin sağlıklı çalışabilmesi için yeterli değildir. Bir yapay sinir ağı modelinde, karesel hatanın mümkün olduğunca sıfıra yakın olması istenir. Hesaplanan hata, problemin lineer bir çözüme sahip olmadığını göstergesidir. Modeli sağlıklı bir şekilde kurabilmek için, çok katmanlı ileri beslemeli bir ağ yapısı kullanılması gerekmektedir.



Şekil 4.2 Tek katmanlı ileri beslemeli ağ modelinde eğitim düşümü

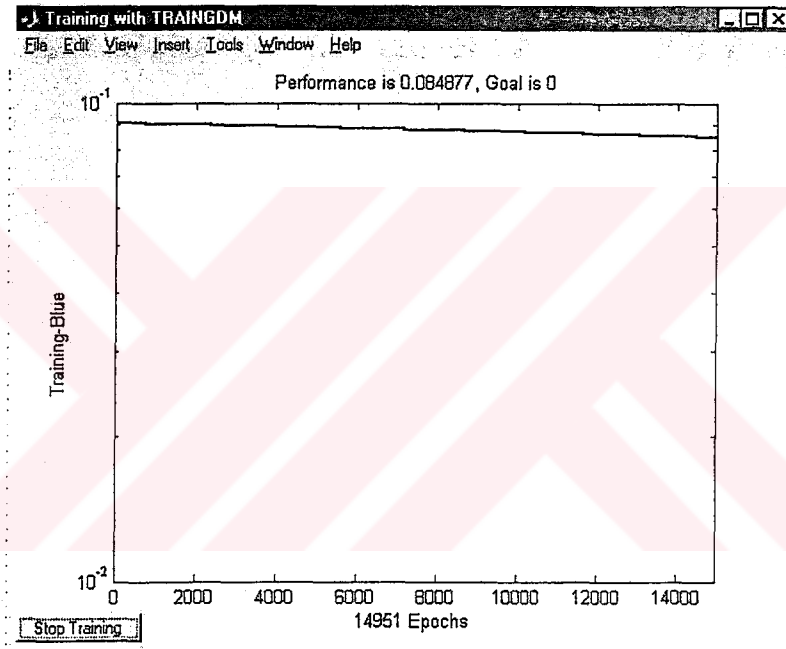
4.3.3 Çok Katmanlı İleri Beslemeli Model

Modelin kurulması sırasında, çok katmanlı ağ modelinin gerekliliği anlaşıldığında, geriye kalan iki soru, modelde kaç katman bulunacağı ve kaç adet saklı nöron kullanılacağıdır. Bunu belirlemek için, basitten komplekse giden bir yöntem izlenmiş ve ilk önce tek saklı katman kullanılmış, saklı nöron sayıları ise teker teker artırılmıştır. Saklı katmanda tek saklı nöron kullanıldığında, elde edilen sonuçta çok fazla bir iyileşme gözlenememiştir ($E = 0.234327$) (Şekil 4.3).



Şekil 4.3 Saklı katmandaki tek nöronla eğitim düşümü

Daha sonra, istenilen gradyan değerine ulaşıncaya kadar, saklı katmandaki nöron sayısı artırılmış ve 16 adet saklı nöron kullanıldığında, tatmin edici bir karesel hata miktarına ulaşılabilmektedir. Şekil 4.4’de görüldüğü gibi karesel hata, 114951 eğitime döngüsü sonucu $E = 0,084877$ değerine kadar azaltılabilmektedir. Bu performans değerinde, hatalı çözülen çatışma sayısı 27 olmaktadır. Şekil 4.2 – 4.4’de görüldüğü gibi, gradyan değeri önceleri düzenli ve hızlı olarak azalmakta, daha sonra azalma miktarı giderek yavaşlayarak, en sonunda neredeyse yatay hale gelmekte, azalma durmaktadır. Şekillerde görülen küçük tepecikler ise yerel minimumları sembolize etmektedir.



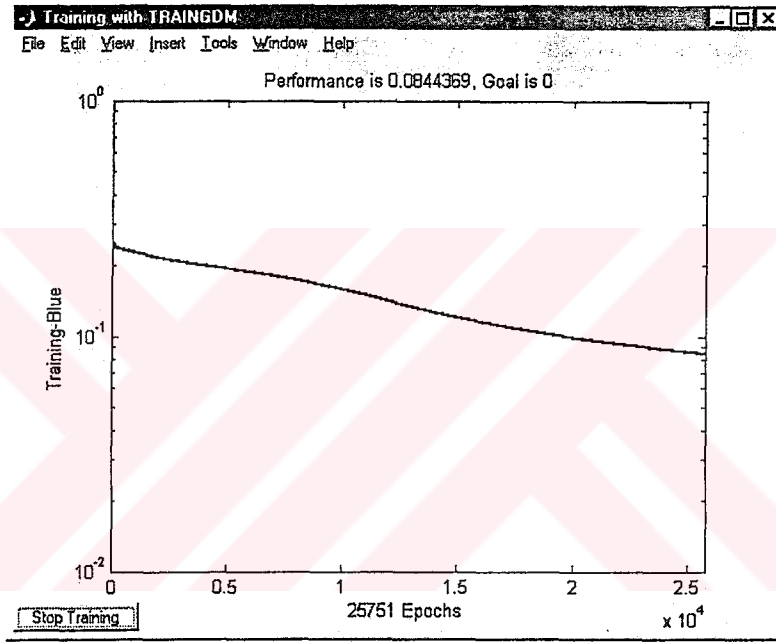
Şekil 4.4 Saklı katmandaki 16 nöronla eğitim düşümü

Daha sonra, saklı katman sayısı artırılarak, sistemin performansında artış olup olmadığı incelenmiştir. Birinci saklı katmanda 16, İkinci saklı katmanda 4 saklı nöron kullanılan modelde, gradyan değeri olarak $E = 0.0847894$ yakalandığında, hatalı çözülen çatışma sayısı 29 olmuştur.

Daha sonra, kritik oranı oluşturan iki değer olan, trenin planlanmış son varış zamanına kalan süre ile kalan en küçük seyir süresi, modele, kritik oranın yerine, iki ayrı giriş parametresi olarak dahil edilmiştir. Böylece sinir ağı yapısı, 10 giriş parametresi ve tek çıkış nöronuna sahip hale gelmiştir. Bu yapıda tek saklı katmanda 20 nöron kullanılmış ve gradyan azaltması

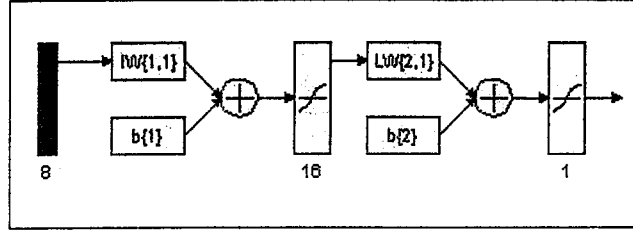
(eğim düşümü) çok daha hızlı gerçekleşmiştir. Ancak, bu yapıda dispeçerin kararları aynı performans değeri ($E = 0.0844369$) için daha hatalı (28 adet) taklit edilebilmiştir (Şekil 4.5).

Her tren için, çatışmanın çözüldüğü istasyon itibarıyla, trenin o ana kadar olan toplam gecikme süresi modele eklenip, performansta herhangi bir iyileşme sağlanıp sağlanmadığı kontrol edilmiştir. Ancak, bu kriter, modelin performansında herhangi bir değişime yol açmamıştır.



Şekil 4.5 10 Girişli modelde saklı katmandaki 20 nöronla eğim düşümü

Şekil 4.6'daki model (8 giriş parametresi, tek saklı katmanda 16 nörona ve tek çıkış nöronuna sahip) tatmin edici sonuçlar sergilediği için, modelde kullanılan karar değişkenleri model çözümüne yaptıkları katkı bakımından incelenmiştir. Her çatışan tren çifti için bu değişkenlerden biri çıkarılarak (elenerek) model test edilmiştir. Ayrıca, kritik oranı iki parametreye bölerek oluşturulan; 10 giriş parametrelili, tek saklı katmandaki 20 nörona ve tek çıkış nöronuna sahip model de benzer şekilde (parametreler tek tek çıkarılarak) test edilmiştir. Yapılan testlerde, saklı nöron sayısı, modele dahil edilen toplam parametre sayısının iki katı alınmıştır. Elde edilen sonuçlar Çizelge 4.2'de görülmektedir.



Şekil 4.6 8 Giriş parametresi, 16 saklı nöron, 1 çıkış nöronuna sahip model

Çizelge 4.2 Modele dahil edilen parametrelerin ağ performansı üzerinde etkileri

Temel Öncelik Sayısı	Kritik Oran	Kalan Min. Süre	Kalan Min. Seyir Süresi	Miyopik Çözümdeki Gecikme	Potansiyel Çatışma Sayısı	Saklı Nöron Sayısı	Hatalı Çözülen Çatışma Sayısı
X	X			X	X	16	27
	X			X	X	12	88
X				X	X	12	59
X	X				X	12	76
X	X			X		12	91
X		X	X	X	X	20	28
X			X	X	X	16	44
X		X		X	X	16	53

X: Modele dahil edilen parametre

4.4 Sayısal Uygulama Sonuçlarına İlişkin Bazı Saptamalar

- Kurulan yapay sinir ağı modelinde, dispeçerin verdiği 331 kararın, 304'ü başarıyla taklit edilebilmiştir (Çizelge 4.2'de satır no 1). Bu da yüzde 92'lik bir başarı oranı demektir. Bu başarı oranına, benzer problemlerle karşılaştırıldığında çok nadir rastlanmaktadır.
- Kritik oranı oluşturan iki değer olan “planlanmış son varış zamanına kalan süre” ile “kalan en küçük seyir süresi”, kritik oranın yerine modele dahil edildiğinde, eğim düşmesi çok daha hızlı gerçekleşmiştir. Ancak, dispeçer kararları daha hatalı taklit edilebilmiştir

Modelde kullanılan bazı çatışma çiftleri Çizelge 4.3'te verilmiştir. Bu çizelgede, çatışan tren çiftlerinden, dispeçerin geçmesine izin verdiği tren üst satıra, beklettiği tren alt satıra

yazılmıştır. Dispeçerin çözümü ve modelin çözümü sütunlarındaki sayılar, sırasıyla modelin vermesi istenen ve modelin verdiği sonuçlardır.

Çizelge 4.3 Trenlerarası çatışma örnekleri ve çözümleri

Çatışan Tren No ¹	Temel Öncelik Sayısı	Kritik Oran	Miyopik Gecikme	Potansiyel Çatışma Sayısı	Dispeçerin Çözümü ²	Modelin Çözümü	Açıklama
11210	50	1.346152	4	1	0	0.8546	Modele aykırı çözüm
11209	50	1.142857	10	2			
11319	50	0.627451	10	2	0	0.9762	Dispeçer geciken trene yol vermiştir
11022	20	1.057692	8	2			
11319	60	0.689076	10	2	0	0.99828	Dispeçer geciken trene yol vermiştir
11210	50	0.8	5	2			
11207	50	1.039216	13	1	0	0.64944	Yol verilen trenin daha fazla gecikmesini önleyen bir karar
11022	20	1.134615	6	4			
11210	50	0.846154	5	0	0	0.72864	Dispeçer gecikmiş trene yol vermiştir
11209	50	1.037594	9	3			
11319	60	0.722689	6	3	0	0.5575	Dispeçer, gecikmesi büyük trene yol vermiştir
11210	50	0.945455	9	2			
11512	70	0.682927	6	3	0	0.53999	Yol verilen trenin daha fazla gecikmesini önleyen bir karar
11207	50	1.06087	12	2			
11021	20	0.901099	5	0	1	0.4921	Yol verilen trenin daha fazla gecikmesini önleyen bir karar
11126	20	1.134615	13	2			

Çizelge 4.3'ün devamı

Çatışan Tren No ¹	Temel Öncelik Sayısı	Kritik Oran	Miyopik Gecikme	Potansiyel Çatışma Sayısı	Dispeçerin Çözümü ²	Modelin Çözümü	Açıklama
11021	20	1.029703	9	1	1	0.44822	Dispeçerle modelin çözümleri farklı
11022	20	1.264706	8	2			
11208	50	1.178571	3	1	1	0.54836	Temel öncelik sayısının önemi
11319	60	0.880952	11	3			
11208	50	1.178571	4	1	0	0.21194	Aykırı olmasına rağmen uyumlu çözüm
11207	50	1.08209	10	4			
11207	50	1.04918	6	3	1	0.89628	Kritik oranın önemi
11210	50	1.072727	9	3			
11207	50	1.04918	11	3	1	0.99455	Miyopik gecikmenin önemi
11210	50	0.981818	4	5			
11209	50	1.086614	6	3	1	0.89328	Potansiyel çatışma sayısı kararda etkili olabilir
11210	50	0.657895	8	1			
11021	20	0.934426	12	4	1	1	Uyumlu bir çözüm
11210	50	1.3	9	0			
11022	20	0.931818	15	2	0	0.03249 1	Uyumlu bir çözüm
11021	20	1.032258	4	1			
12517	80	0.408163	11	4	0	0.00629 09	Uyumlu bir çözüm
11208	50	1.168675	2	3			

Çizelge 4.3'ün devamı

Çatışan Tren No ¹	Temel Öncelik Sayısı	Kritik Oran	Miyopik Gecikme	Potansiyel Çatışma Sayısı	Dispeçerin Çözümü ²	Modelin Çözümü	Açıklama
11022	20	1.128205	12	5	1	0.99995	Uyumlu bir çözüm
14005	100	1.138686	4	1			
11210	50	1.01087	8	4	1	0.91306	Uyumlu bir çözüm
14005	100	0.878924	16	5			
11207	50	0.676471	9	0	0	0.08512 1	Uyumlu bir çözüm
11126	20	1.134615	9	2			
11021	20	0.732673	14	0	1	0.88377	Uyumlu bir çözüm
11126	20	1.05	3	0			

¹Çatışan tren çiftlerinden, dispeçerin geçmesine izin verdiği tren üst satıra, beklettiği tren alt satıra yazılmıştır.

²Dispeçerin çözümü 1 ise, küçük numaralı tren geçiyor, büyük numaralı tren bekletiliyor; aksi durumda tersi geçerlidir.

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, trenlerin önceden hazırlanmış çizelgeden sapması durumunda, ortaya çıkabilecek trenlerarası çatışmalara çözüm üreten dispeçerlerin, bu kararları nasıl (hangi parametreleri dikkate alarak) aldığını inceleyen ve daha sonra bu kararları taklit eden bir sistem geliştirilmiştir. Bu sistem için, yapay sinir ağı tekniği kullanılmıştır. Demiryolu trafik yönetimi problemi uzun bir süredir araştırmacıların ilgisini çekmesine rağmen, yapılan çalışmalarda trenlerarası çatışmalar için optimum çözümü bulunamamış, bulunan çözümler ise problemin çok büyük boyutlu ve karmaşık olmasından dolayı optimuma yakın olmaktan öteye gidememiştir. Çeşitli kontrol problemlerinde kontrolör kararlarının taklit edilmesinde, yüzde 100'lük bir başarı elde edilememiştir. Burada sunulan modelde, yapay sinir ağı tekniği kullanılarak, dispeçer kararları yüzde 92 başarı ile taklit edilebilmiştir. Sistemin getirdiği bazı yenilikler şöyledir:

- Yapay sinir ağı modeli, ilk olarak dispeçerin kararlarını nasıl aldığını inceleyip, kullanılan karar değişkenlerinin önceliklerini (ağırlıklarını) belirler. Ardından modele tanıtılan örneklerdeki çözümleri taklit ederek, gelecekteki çatışmalar için çözümler üretir.
- Yapay sinir ağlarına, yeni işletme koşulları kolaylıkla tanıtılabilmekte, güncellenen verilerle yeni koşullar modellenebilmektedir.
- Sunulan model, ikiden fazla trenin aynı anda aynı istasyon/saydingde bulunması durumunda, bunları ikili olarak inceleyip, karar verme yeteneğine sahiptir.
- Kullanılan model, karar değişkenleri birbirine çok yakın değerler olsa bile, kesin karar verebilecek şekilde geliştirilmiştir.
- Modelin, dispeçerin kararlarına uygun olmayan sonuçlar verdiği durumlarda, dispeçerin modelde kullanılmayan bazı kriterleri göz önünde bulundurduğu düşünülebilir. Bu tür

durumlarda, daha uygun sonuçlar elde edebilmek için, başka parametrelerin de modele dahil edilmesi gerekmektedir.

Bununlar birlikte, algoritma aşağıdaki öneriler doğrultusunda geliştirilebilir:

- Algoritma, (iki yönlü tren hareketlerine izin verilen sinyalizasyonlu) çift ve çok hatlı demiryolu kesimlerinde çatışma çözümlerini de yapabilecek şekilde genişletilebilir.
- Algoritma, karşılaşma çatışmasının yanında öne geçme çatışmalarını da çözebilecek şekilde eğitilebilir.
- İki'den çok trenin çatışması (buluşması) durumu için, yeni parametreler eklenerek modelin çözüm kabiliyeti geliştirilebilir
- Farklı hatlardan çok sayıda örnek alınarak, model her yerde yüksek performansla çalışacak şekilde genişletilebilir.
- Geliştirilen model, bir karar destekleyici sistem kapsamında, deneyimsiz dispeçerlerin yetiştirilmesi esnasında eğitime amaçlı bir araç olarak kullanılabilir.
- Böyle bir karar destekleyici sistem, kullanıcı dostu bir paket program olarak hazırlandığında, çevrim dışı kullanım için dispeçerlerin hizmetine sunulabilir.

KAYNAKLAR

- Cordeau, J-F., Toth, P. ve Vigo, D. (1998) A survey of Optimization Models for Train Routing and Scheduling, *Transportation Science*, Vol. 32, No. 4, s. 380-404.
- Higgins, A., Kozan, E. ve Ferreira, L. (1997) Heuristic Techniques for Single Line Train Scheduling, *Journal of Heuristics*, 3, s. 43-62
- Lin, C.T. ve George Lee, C.S. (1996) *Neural Fuzzy Systems; A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall.
- Malavasi, G. ve Ricci, S. (2001) Simulation of Stochastic Elements in Railway Systems Using Self-Learning Processes, *European Journal of Operational Research*, 131, s. 262-272.
- Potvin, J-Y., Shen, Y., Dufour, G. ve Rousseau J-M. (1995) Learning Techniques for an Expert Vehicle Dispatching System, *Expert Systems With Applications*, Vol. 8, No.1, s. 101-109.
- Şahin, İ. (1996) *Trenlerarası Çatışma Yönetimine Dayalı Trafik Kontrolü İçin Bir Karar Destekleyici Sistem*, Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul
- Şahin, İ. (1999) *Railway Traffic Control and Train Scheduling Based on Inter-Train Conflict Management*, *Transportation Research Part B*, 33, s. 511-534.
- TCDD (1995) *Haydarpaşa-Ankara ve Arifiye-Adapazarı Kısımları Trenlerinin Kalkış Varışlarını Gösterir Tarife*.
- TCDD (2000) *Haydarpaşa-Ankara ve Arifiye-Adapazarı Kısımları Trenlerinin Kalkış Varışlarını Gösterir Ek Tarife*.
- Teodorovic, D. ve Vukadinovic K. (1996) *Traffic Control and Transport Planning: A Fuzzy Sets and Neural Network Approach*, Kluwer Academic Publishers.
- Yıldırım, T. (2002) *Yapay Sinir Ağları Ders Notları*, yayımlanmamış çalışma, Yıldız Teknik Üniversitesi, Elektronik ve Haberleşme Bölümü.

İNTERNET KAYNAKLARI

- [1] www.generation5.org

ÖZGEÇMİŞ

Doğum tarihi	17.05.1979	
Doğum yeri	Fatih, İstanbul	
Lise	1989-1996	Özel Kültür Lisesi
Lisans	1996-2000	Yıldız Teknik Üniversitesi, İnşaat Fakültesi, İnşaat Mühendisliği Bölümü
Yüksek Lisans	2001-2003	Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İnşaat Müh. Anabilim Dalı, Ulaştırma Programı

Çalıştığı kurumlar

2001

Özgür Yapı Taahhüt, Sanayii ve Ticaret Ltd. Şti.