

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**



**EKSTREMUM PROBLEMLERİ İÇİN GRADYANT TABANLI
YAKLAŞIM**

YÜKSEK LİSANS TEZİ

PERVİN ÇALIK

BALIKESİR, HAZİRAN - 2014

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**



**EKSTREMUM PROBLEMLERİ İÇİN GRADYANT TABANLI
YAKLAŞIM**

YÜKSEK LISANS TEZİ

PERVİN ÇALIK

BALIKESİR, HAZİRAN - 2014

KABUL VE ONAY SAYFASI

Pervin ÇALIK tarafından hazırlanan "EKSTREMUM PROBLEMLERİ İÇİN GRADYANT TABANLI YAKLAŞIM" adlı tez çalışmasının savunma sınavı 10.06.2014 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Danışman

Yrd. Doç. Dr. Fırat EVİRGEN

Üye

Yrd. Doç. Dr. Aslan Deniz KARAOĞLAN

Üye

Yrd. Doç. Dr. Beyza Billur İSKENDER


.....

.....

.....

Jüri üyeleri tarafından kabul edilmiş olan bu tez BAÜ Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Cihan ÖZGÜR

.....

ÖZET

**EKSTREMUM PROBLEMLERİ İÇİN GRADYANT TABANLI YAKLAŞIM
YÜKSEK LİSANS TEZİ
PERVİN ÇALIK
BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI**

(TEZ DANIŞMANI: YRD. DOÇ. DR. FIRAT EVİRGEN)

BALIKESİR, HAZİRAN - 2014

Optimizasyon verilen koşullar altında en iyi sonucun elde edilmesi işlemidir. Optimizasyonun temel bilimlerde ve mühendislik alanlarında birçok uygulaması mevcuttur. Bu uygulama alanlarındaki bir çok problemin çözümü çok zor hatta bazen imkansızdır. Fakat son zamanlarda matematik ve bilgisayardaki gelişmelerle birlikte birçok teknik geliştirilerek bu problemlerin çözümü araştırılmıştır. Bu tekniklerden bazıları gradyant tabanlı metodlar ve yapay sinir ağları (YSA)'dır. Özellikle YSA'lar insan beyni gibi tasarlandığı için optimizasyon problemlerinin çözümlerinde önemli bir popülerliğe sahiptir.

Bu tezde ilk olarak optimizasyon ve YSA'lar hakkında genel bilgiler verilmiş olup, sonrasında da optimizasyon ve YSA'ların ortak kullanımı hakkında bilgiye yer verilmiştir.

Son olarak, tezi son iki bölümünde, değiştirilmiş bariyer-genişletilmiş Lagrange metodu ve Hiperbolik ceza metodu hakkında temel tanım ve teoremlere ve sonrasında bu yöntemler ile modellenen gradyant tabanlı dinamik sistem yaklaşımına yer verilmiştir. Ayrıca bu yaklaşım ile bazı test problemleri modellenerek çözülmüştür. Bu yaklaşım YSA'lara adapte edilebilir.

ANAHTAR KELİMELER: Optimizasyon Problemleri, YSA'lar, Hiperbolik Ceza Metodu, Değiştirilmiş Bariyer-Genişletilmiş Lagrange Metodu

ABSTRACT

GRADIENT BASED APPROACH FOR EXTREMUM PROBLEMS

MSC THESIS

PERVIN ÇALIK

BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE

MATHEMATICS

(SUPERVISOR: ASSIST. PROF. DR. FIRAT EVİRGEN)

BALIKESİR, JUNE 2014

Optimization is a process that is aimed to obtain best results under the given conditions. Optimization has many applications area in fundamental sciences and engineering. Many problems in these areas are hard to solve, sometimes impossible. But, in the recent times by the improvements in the mathematics and computers, many techniques are developed to solve these kinds of problems. Some of these are gradient based methods and Artificial Neural Network's (ANN's). Especially ANN's have significant popularity in solving optimization problems for designing like human brain.

Firstly, in this thesis some fundamental definitions and theorems about optimization theory and some literature about ANN's are given. In the following, we are given the connection between these two areas.

Finally, in the last two chapters of the thesis, we are mentioned fundamental definitions and theorems about Modified Barrier and Augmented Lagrangian method and Hyperbolic Penalty method and then the gradient based dynamical system approach which was modeled by these methods is given. Furthermore some test problems are modeled and solved with this approach. This approach can be adapted to the ANN's.

KEYWORDS: Optimization problems, ANN's, hyperbolic penalty method, modified barrier and augmented Lagrangian method.

İÇİNDEKİLER

Sayfa

ÖZET.....	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ	iv
TABLO LİSTESİ	v
SEMBOL LİSTESİ	vi
KISALTMALAR LİSTESİ.....	vii
ÖNSÖZ.....	viii
1. GİRİŞ.....	1
2. OPTİMİZASYON	4
2.1 Eşitlik Kısıtlı Optimizasyon Problemleri	4
2.1.1 Teğet ve Normal Uzaylar.....	5
2.1.2 Lagrange Koşulları	7
2.2 Eşitsizlik Kısıtlı Optimizasyon Problemleri.....	11
3. YAPAY SİNİR AĞLARI.....	15
3.1 YSA'nın Yapısı	15
3.2 Yapay Sinir Hücresi (Proses Elemanı).....	17
3.2.1 Yapay Sinir Ağlarının Özellikleri.....	20
3.2.2 YSA'ların Tarihçesi	22
3.2.3 YSA'ların Kullanıldığı Alanlar	22
4. OPTİMİZASYON VE YAPAY SİNİR AĞLARI.....	25
4.1 Ceza (Penalty) Fonksiyonu Metodları.....	26
4.1.1 Doğrusal Program (LP) Problemleri.....	26
4.1.2 Doğrusal Olmayan Programlama (NLP) Modelleri.....	28
4.2 Lagrange Çarpanları Metodu.....	29
4.2.1 Doğrusal Programlama (LP) Metodu	29
4.2.2 Doğrusal Olmayan Program (NLP) Problemleri	30
5. DEĞİŞTİRİLMİŞ BARIYER VE GENİŞLETİLMİŞ LAGRANGE METODU İLE GRADYANT TABANLI YAKLAŞIM.....	32
5.1 Problemin İfadesi ve Temel Varsayımları.....	32
5.2 Değiştirilmiş Bariyer ve Genişletilmiş Lagrange Fonksiyonu	34
5.3 Gradyant Tabanlı Yaklaşım.....	35
6. HİPERBOLİK CEZA METODU İLE GRADYANT TABANLI YAKLAŞIM	38
6.1 Hiperbolik Ceza.....	39
6.2 Hiperbolik Ceza Algoritması.....	39
6.3 Gradyant Tabanlı Yaklaşım.....	40
7. SONUÇLAR.....	44
8. KAYNAKLAR.....	45

ŞEKİL LİSTESİ

Sayfa

Şekil 3.1: Yapay Sinir Hücresi	15
Şekil 3.2: Yapay Sinir Ağının Yapısı	16
Şekil 3.3: Yapay Sinir Hücresinin Yapısı	17
Şekil 3.4: Doğrusal Aktivasyon Fonksiyonu Grafiği	18
Şekil 3.5: Adım Aktivasyon Fonksiyonu Grafiği	18
Şekil 3.6: Sigmoid Aktivasyon Fonksiyonu Grafiği	19
Şekil 3.7: Tanjant Hiperbolik Aktivasyon Grafiği	20
Şekil 3.8: Yapay Sinir Ağı Türleri [23]	23
Şekil 5.1: Örnek 5.3.1 de $x(t)$ 'nin çözüm yörüngesi	37
Şekil 6.1: Örnek 6.3.1 de $x(t)$ 'nin çözüm yörüngesi	42
Şekil 6.2: Örnek 6.3.2 de $x(t)$ 'nin çözüm yörüngesi	43

TABLO LİSTESİ

Sayfa

Tablo 4.1: Optimizasyon Problemleri için Ceza Fonksiyonu Metodları	29
Tablo 4.2: Optimizasyon Problemleri için Lagrange Çarpanları Metodları	31

SEMBOL LİSTESİ

- $f(x)$: Amaç fonksiyonu
- $h_i(x)$: i . eşitlik kısıtı
- $g_j(x)$: j . eşitsizlik kısıtı
- $T(x^*)$: S yüzeyi üzerinde x^* noktasındaki tanjant uzayı
- $\mathcal{N}(x^*)$: S uzayı üzerinde x^* noktasındaki normal uzay
- $l(x, \lambda)$: Lagrange fonksiyonu
- $J(x^*)$: Aktif eşitsizlik kısıtlarının indis kümesi
- $\dot{x}(t) = \frac{dx}{dt}$: $x(t)$ 'nin t bağımsız değişkenine göre türevi
- S : Kısıtların tanımlandığı yüzey
- $F(x, u, v, k)$: Değiştirilmiş bariyer ve genişletilmiş Lagrange fonksiyonu
- $P(y, \lambda, \tau)$: Hiperbolik ceza metodunun ceza fonksiyonu
- $F(x, \lambda, \tau)$: Hiperbolik ceza fonksiyonu

KISALTMALAR LİSTESİ

KKT	:	Karush-Kuhn-Tucker
YSA	:	Yapay Sinir Ağları
LP	:	Doğrusal Programlama
NLP	:	Doğrusal Olmayan Programlama
DBGL	:	Değiştirilmiş Bariyer ve Genişletilmiş Lagrange

ÖNSÖZ

Bu çalışmam süresince bilgi ve deneyimleriyle yol gösterip, her türlü yardım ve fedakarlığı sağlayan, değerli hocam ve çalışmamın yöneticisi Yrd. Doç. Dr. Fırat EVİRGEN'e;

Küçük dünyasında bana göstermiş olduğu kocaman anlayışı için biricik kızım Meryem Berru ÇALIK'a;

Çalışmamın başından sonuna kadar sonsuz anlayışları, sabırları ve her türlü destekleri için sevgili aileme ve biricik kocam Zafer Recep ÇALIK'a sonsuz teşekkürlerimi sunarım.

1. GİRİŞ

İnsanoğlunun bir işi en iyi şekilde gerçekleştirme isteği geçmişten günümüze değişmeden gelen en büyük amaçlarından biridir. Değişen tek şey bu en iyiye ulaşma yöntemleridir. Matematik ve bilgisayardaki gelişmeler günlük hayatta karşılaştığımız problemleri matematiksel olarak ifade edip çözerek bu çözümleri gerçek hayatta kullanma olanağı vermiştir. Eski uygarlıklara bakacak olursak; Yunan tarihçisi Herodotus'a göre, Mısırlılar Nil nehrinin her yıl taşması sonucu arazi sınırlarının yeniden belirlenmesi ve yeni sınırlara göre vergilendirme işleminin en iyi yolla yapılabilmesi için çaba sarf etmişlerdir. Bu çabalar, ölçme ve karar verme aracı olarak düzlem geometrisinin temel kavramlarının oluşturulmasına yol açmıştır [1]. Mısırlılar, Nil nehrinin bahar dönemlerindeki yıllık taşmalarında nehir kıyısından toplu halde uzaklaşıp sular çekildiğinde yine toplu halde geri dönüyorlardı. Çekilme işlemi uzun sürede gerçekleştiği için günlerce önceden halk uyarılmalıydı. Bu amaçla, Mısırlılar en iyi çekilme zamanını hesaplayabilmek için bir tür takvim bile geliştirmişlerdi. Söz konusu takvimi de sayma ve geometri konusundaki birikimlerini kullanarak yapmışlardı [2,3].

Newton ve Leibniz tarafından Kalkülüs'ün (Calculus) 17. yüzyılda geliştirilmesi optimizasyon teorisinin gelişiminde önemli yere sahip olmuştur. Kalkülüs, hem matematiksel bir fonksiyonun hem de fonksiyon oluşturabilen bağımsız değişkenlerin maksimum veya minimum cinsinden optimal koşullarının elde edilmesine olanak sağlamaktadır. Kalkülüs'ün kullanımı düzgün-davranışlı fonksiyonlarla sınırlandırılmıştır. Ancak, Kalkülüs uygulamalarında karşılaşılan cebirsel problemlerin çözümü bazen güç olabilmektedir. Dolayısıyla, Kalkülüs pragmatik anlamda gerçek dünya problemlerinin optimizasyonunda yeterli ve güçlü bir araç olamamaktadır [1].

J.L. Lagrange'ın 1788 yılında Lagrange çarpanları yöntemini bilim dünyasının sunması önemli bir adım olmuştur. Daha sonraları 1939'da W. Karush'un kısıtlandırılmış problemler için optimallik koşullarını bulması optimizasyon teorisinde yeni bir kapı açmıştır. II. Dünya Savaşı'nın başlamasıyla 1942'de İngiltere

ve Amerika Birleşik Devletleri'nin Yöneylem Araştırması gruplarını oluşturması optimizasyon dünyası için bir dönüm noktası olmuştur. Sezgisel optimizasyon araçlarından olan yapay sinir ağları 1943 yılında, bir sinir hekimi olan Warren McCulloch ile bir matematikçi olan Walter Pitts tarafından gerçekleştirilmiştir[4].

II. Dünya Savaşı'ndan sonra yeni sınıf optimizasyon teknikleri geliştirildi. Söz konusu teknikler daha karmaşık problemlere başarıyla uygulandı. Bunda, yüksek hızlı dijital bilgisayarların geliştirilmesi ve optimum değerlerin elde edilmesi için nümerik tekniklere matematiksel analizin uygulanması son derece etkili olmuştur. Nümerik teknikler Kalkülüs'ün bir takım zorluklarını ortadan kaldırmıştır [1].

Lineer programların çözümü için Simplex yöntemi 1947'de G.B. Dantzig tarafından geliştirildi. R. Bellman 1950'de dinamik programlama modelini ve çözümünü geliştirdi. 1951'de H. Kuhn ve A. Tucker daha önce Karush'un önerdiği kısıtlandırılmış problemler için optimallik koşullarını tekrar formüle ederek doğrusal olmayan programlama modelleri üzerinde çalıştılar. Aynı yıl, J. Von Neumann, G. Dantzig ve A. Tucker primal-dual lineer programlama modellerini geliştirdi. Kuadratik programlama 1956'da M. Frank ve P. Wolfe tarafından geliştirildi. 1958'deki önemli bir katkı R. Gomory tarafından tamsayılı programlama olarak adlandırıldı. A. Charnes ve W. Cooper şans kısıtlı programlama modellerini 1959'da optimizasyon dünyasına armağan ettiler. 1960'da sezgisel optimizasyon araçlarından birisi olan yapay zeka ve yöneylem araştırması ilişkilerini içeren çalışmalar yapıldı. Hedef programlama modeli yine A. Charnes ve W. Cooper tarafından 1965 yılında geliştirildi. 1975'de çok amaçlı karar verme teorisinin temelleri M. Zeleny, S. Zionts, J. Wallenius, W. Edwards ve B. Roy tarafından atıldı. L. Khachian lineer programlama modellerinin çözümü için farklı bir algoritma olan elips yöntemini 1979'da geliştirdi. 1984'te, N. Karmarkar lineer programlama için alternatif bir çözüm algoritması olan iç nokta algoritmasını geliştirdi. 1992'de J.H. Holland tarafından bir sezgisel optimizasyon tekniği olarak kabul edilen genetik algoritma geliştirildi [4,5].

Biyolojik sınırlardan esinlenerek elde edilen YSA, doğrusal olmayan ve paralel bilgi işleme özellikleriyle; bulunduğu ortamın değişmesiyle cevaptaki davranışı değiştirebilme, giriş uyarılarındaki küçük değişimleri tolere edebilme, değişik bazı uyarılar karşısında daha önceki uyarılar arasında benzer özellikler

keşfederek deneyimi olmadığı halde uyarıyı cevaplayabilme gibi üstün özelliklere sahiptir [6,7].

Optimizasyon çalışmalarında yapay sinir ağları en güçlü teknikler arasında yer alır. Veri madenciliği, optik karakter taşıma, optimum rota belirleme, parmak izi tanıma, malzeme analizi, iş çizelgelemesi ve kalite kontrol, tıbbi analiz gibi birçok alanda günlük hayatımızda göreceğimiz başarılı örneklerine rastlamak mümkündür [8].

Günümüzde optimum çözümleri için hala diferansiyel hesap yöntemleri kullanılmaktadır. Optimizasyon modelleme geleneksel olarak matematik programlama olarak adlandırılmaktadır [9]. Diğer bir ifadeyle, matematik programlama, optimizasyon modelinin kurulması ve çözümün elde edilmesi işlemine verilen genel isimdir. Geçmişten gelen bir gelenekle günümüzde de "matematik programlama" ve "optimizasyon" kavramları eşanlamlı olarak kullanılmaktadır [4]

2. OPTİMİZASYON

En genel anlamıyla optimizasyon en iyileme anlamına gelmektedir. Optimizasyon hedeflenen amacı minimum ya da maksimum yapacak koşulları bulma işlemi olarak tanımlanabilir.

2.1 Eşitlik Kısıtlı Optimizasyon Problemleri

Eşitlik kısıtlı optimizasyon problemleri aşağıdaki gibi tanımlanır

$$\begin{aligned} \min f(x) &= f(x_1, x_2, \dots, x_n) \\ \text{st. } h_i(x) &= 0 \quad (i=1, 2, \dots, n) \text{ koşulu ile} \end{aligned} \quad (2.1)$$

$x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ tasarım vektörü olarak söylenen n boyutlu bir vektördür. $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ amaç fonksiyonunu ve $h_i(x): \mathbb{R}^n \rightarrow \mathbb{R}$ eşitlik kısıtlarını temsil eder.

Tanım 2.1.1: Eğer $h_1(x^*) = 0, h_2(x^*) = 0, \dots, h_m(x^*) = 0$ eşitlik kısıtlarını sağlayan bir x^* noktası için $\nabla h_1, \nabla h_2, \dots, \nabla h_m$ gradyant vektörleri lineer bağımsız ise bu noktaya düzgün (regüler) nokta denir.

$Dh(x^*), \quad h = [h_1, h_2, \dots, h_m]^T$ matrisinin Jakobiyen matrisi olmak üzere

$$Dh(x^*) = \begin{pmatrix} Dh_1(x^*) \\ \cdot \\ \cdot \\ \cdot \\ Dh_m(x^*) \end{pmatrix} = \begin{pmatrix} \nabla h_1(x^*)^T \\ \cdot \\ \cdot \\ \cdot \\ \nabla h_m(x^*)^T \end{pmatrix} \text{ sağlanır.}$$

x^* noktasının düzgün bir nokta olabilmesi için gerek ve yeter şart $\text{rank}(Dh(x^*)) = m$ olmasıdır ($m \leq n$).

$h_1(x) = 0, h_2(x) = 0, \dots, h_m(x) = 0, h_i: \mathbb{R}^n \rightarrow \mathbb{R}$ eşitlik kısıtlarının kümesi olmak

üzere $S = \{x \in \mathbb{R}^n, h_1(x) = 0, h_2(x) = 0, \dots, h_m(x) = 0\}$ şeklinde bir yüzey tanımlanır. S yüzeyindeki noktaların düzgün bir nokta olduğu kabul edersek yüzeyin boyutu $n - m$ ' dir.

$$\dim S = \dim \{x : h_i(x) = 0\} = n - m.$$

2.1.1 Teğet ve Normal Uzaylar

Tanım 2.1.1.1: Herhangi bir S yüzeyinde C eğrisi $t \in (a, b)$ ile sürekli parametrize olan $\{x(t) \in S : t \in (a, b)\}$ noktalar kümesi ile tanımlanır. C eğrisi bir x noktası ile S yüzeyinde dolaşan bir yol olarak kabul edilebilir.

Tanım 2.1.1.2: Eğer $\forall t \in (a, b)$ için

$$\dot{x}(t) = \frac{dx}{dt} = \begin{pmatrix} \dot{x}_1(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n(t) \end{pmatrix}$$

var ise $C = \{x(t) : t \in (a, b)\}$ eğrisi diferansiyellenebilir denir.

Benzer şekilde;

$$\ddot{x}(t) = \frac{d^2x}{dt^2} = \begin{pmatrix} \ddot{x}_1(t) \\ \cdot \\ \cdot \\ \cdot \\ \ddot{x}_n(t) \end{pmatrix}$$

sağlanırsa ikinci mertebeden diferansiyellenebilir denir. Burada $\dot{x}(t)$ ve $\ddot{x}(t)$ vektörleri sırasıyla t zamanında $x(t)$ pozisyonunda bulunan ve C eğrisi üzerinde hareket eden bir noktanın hızı ve ivmesi olarak görülebilir. $\dot{x}(t)$ vektörü, $x(t)$

noktasında C eğrisine teğettir denir. Dolayısıyla içinde bulunduğu S yüzeyine de teğettir.

Tanım 2.1.1.3: $S = \{x \in \mathbb{R}^n : h(x) = 0\}$ yüzeyi üzerinde bir x^* noktasındaki tanjant uzayı $T(x^*) = \{y : Dh(x^*)y = 0\}$ kümesi ile tanımlanır. (Teğetlerin oluşturduğu uzay tanjant uzayıdır). Tanım gereği $T(x^*)$ tanjant uzayı $Dh(x^*)$ matrisinin sıfır uzayı (null space) olarak görülebilir. Yani;

$$T(x^*) = \mathcal{N}(Dh(x^*)) \text{ dir.}$$

Tanjant uzayı \mathbb{R}^n 'nin bir alt uzayıdır.

Tanım 2.1.1.4: $S = \{x \in \mathbb{R}^n : h(x) = 0\}$ yüzey üzerinde bir x^* noktasında $\mathcal{N}(x^*)$ normal uzayı

$$\mathcal{N}(x^*) = \left\{ x \in \mathbb{R}^n : x = Dh(x^*)^T z, z \in \mathbb{R}^m \right\} \text{ kümesi ile tanımlanır.}$$

$\mathcal{N}(x^*)$, \mathbb{R}^n 'in $\nabla h_1(x^*), \nabla h_2(x^*), \dots, \nabla h_m(x^*)$ vektörleri ile gerilen bir alt uzayıdır.

$$\begin{aligned} \mathcal{N}(x^*) &= \text{span} \left\{ \nabla h_1(x^*), \nabla h_2(x^*), \dots, \nabla h_m(x^*) \right\} \\ &= \left\{ x \in \mathbb{R}^n : x = z_1 \nabla h_1(x^*) + \dots + z_m \nabla h_m(x^*), z_1, z_2, \dots, z_m \in \mathbb{R} \right\}. \end{aligned}$$

Lemma 2.1.1.5: $T(x^*) = \mathcal{N}(x^*)^\perp$ ve $T(x^*)^\perp = \mathcal{N}(x^*)$ dır [10].

2.1.2 Lagrange Koşulları

Bu kısımda eşitlik kısıtlarına sahip optimizasyon problemleri için birinci mertebeden gerek koşullar verilecektir. Bu sonuçlar Lagrange teoremi olarak bilinir. Teoremin daha iyi anlaşılabilmesi için iki değişkenli kısıtlı bir probleme bakılacaktır [10].

$h : \mathbb{R}^2 \rightarrow \mathbb{R}$ bir kısıt fonksiyonu olsun. Her bir x noktasındaki $\nabla h(x)$ gradyant vektörü bu noktadan geçen seviye kümesine dik olmalıdır. Şimdi $x^* = [x_1^*, x_2^*]^T$ için $h(x^*) = 0$ ve $\nabla h(x^*) \neq 0$ olarak alalım. x^* noktasındaki seviye kümesi $\{x : h(x) = 0\}$ kümesi ile tanımlanır. Bu seviye kümesini x^* 'ın bir komşuluğunda sürekli diferansiyellenebilir $\{x(t)\}$, $x : \mathbb{R} \rightarrow \mathbb{R}^2$ eğrisi ile parametrize edilir. Yani

$$x(t) = \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}, t \in (a, b), x^* = x(t^*), \dot{x}(t^*) \neq 0, t^* \in (a, b) \text{ olsun. } \nabla h(x^*) \text{'in } \dot{x}(t^*)$$

'a dik olduğunu gösterelim. $\forall t \in (a, b)$ için

$$\begin{aligned} h(x(t)) = 0 &\Rightarrow \frac{d}{dt} h(x(t)) = 0 \\ &\Rightarrow \nabla h(x(t))^T \dot{x}(t) = 0 \end{aligned}$$

Dolayısıyla $\nabla h(x^*)$, $\dot{x}(t^*)$ 'a diktir. Şimdi x^* noktasının $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ fonksiyonunun $\{x : h(x) = 0\}$ kümesi üzerinde minimum olduğunu Kabul edelim. Burada $\nabla f(x^*)$ 'ın da $\dot{x}(t^*)$ 'a dik olduğunu gösterelim. Bunun için t^* noktasında minimum olan $\phi(t) = f(x(t))$ bileşke fonksiyonunu ele alalım. x^* minimum olduğundan birinci mertebeden gerek koşul gereği

$$0 = \frac{d}{dt} \Big|_{t^*} = \nabla f(x(t^*)) \dot{x}(t^*) = \nabla f(x^*)^T \dot{x}(t^*)$$

olur. Buradan da $\nabla f(x^*)$ 'in $\dot{x}(t^*)$ 'a dik olduğu görülür. Böylece $\nabla h(x^*)$ da $\dot{x}(t^*)$ dik olduğundan $\nabla f(x^*)$ ile $\nabla h(x^*)$ paralel, yani $\nabla f(x^*)$, $\nabla h(x^*)$ 'in bir skaler katıdır [10].

Teorem 2.1.2.1 (Lagrange koşulları):

x^* noktası $h(x)=0$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$, ($m \leq n$) olacak şekilde $f: \mathbb{R}^n \rightarrow \mathbb{R}$ fonksiyonu bir lokal minimum olsun. Kabul edelim ki x^* regüler bir nokta olsun. Bu durumda

$$\nabla f(x^*) + \lambda^* \nabla h(x^*) = 0$$

Olacak şekilde bir $\lambda^* \in \mathbb{R}^m$ vardır [10].

İspat: Burada $\exists \lambda^* \in \mathbb{R}^m$ için

$$\nabla f(x^*) = -Dh(x^*)^T \lambda^*$$

olduğunu göstermemiz gerekir. Yani $\nabla f(x^*)$ 'ın $\mathcal{N}(x^*)$ 'in bir elemanı olduğunu göstermeliyiz.

Lemma 2.1.1.5 'den $\nabla f(x^*) \in T(x^*)^\perp$ olmalıdır. $y \in T(x^*)$ olarak alalım. Dolayısıyla $\forall t \in (a, b)$ için $h(x(t))=0$ ve $\exists t^* \in (a, b)$ için $x(t^*)=x^*$, $\dot{x}(t^*)=y$ olan bir $\{x(t): t \in (a, b)\}$ diferansiyellenebilen bir eğri vardır. Şimdi $\phi(t) = f(x(t))$ bileşke fonksiyonunu ele alalım. t^* lokal minimum olduğundan birinci mertebeden gerek koşuldan

$$\frac{d\phi}{dt}(t^*) = 0$$

olmalıdır. Zincir kuralını kullanarak;

$$\frac{d\phi}{dt}(t^*) = Df(x^*)\dot{x}(t^*) = Df(x^*)y = \nabla f(x^*)^T y = 0$$

elde edilir. $\forall y \in T$ için $\nabla f(x^*)^T y = 0$ sağlandığından $\nabla f(x^*) \subset T(x^*)^\perp$ dir.

Dolayısıyla $\nabla f(x^*)$ normalinde elemanıdır.

Lagrange teoremine göre, x^* bir ekstremum nokta ise amaç fonksiyonunun $\nabla f(x)$ gradyantı kısıtların gradyantının bir lineer birleşimi şeklinde yazılabilir. Lagrange koşulları eşitlik kısıtlarına sahip optimizasyon problemlerinin minimumları için birinci mertebeden gerek koşuldur. Dikkat edilirse Lagrange koşulları sadece gerek koşullardır. Yani olası ekstremum noktaları bulur. Bunlar maksimum ya da minimum olmayabilir [10].

Lagrange fonksiyonu;

$$l: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$$

$$l(x, \lambda) \triangleq f(x) + \lambda^T h(x)$$

olarak tanımlanır. Lagrange fonksiyonu x ve λ 'a göre türevlerini sıfıra eşitlenmesiyle

$$Dl(x^*, \lambda^*) = (D_x l(x^*, \lambda^*), D_\lambda l(x^*, \lambda^*)) = 0^T$$

x^* lokal minimumluğu için bir gerek şart oluşturur. Yani

$$\begin{aligned} \nabla f(x^*) + \lambda \nabla h(x^*) &= 0 \\ h(x^*) &= 0 \end{aligned}$$

Bu Lagrange koşulları eşitlik kısıtlarına sahip optimizasyon problemlerinin minimumları için birinci mertebeden gerek koşuldur [10].

İkinci Mertebeden Koşullar (Second Order Conditions):

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ve $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$ iki kez diferansiyellenebilir fonksiyonlar, yani $f, h \in C^2$ olmak üzere

$$l(x, \lambda) = f(x) + \lambda^T h(x)$$

Lagrange fonksiyonu olmak üzere $D^2l(x, \lambda)$ ile Lagrange fonksiyonunun x 'e göre Hesse matrisini ele alalım.

Teorem 2.1.2.2 (ikinci Mertebeden Gerek Koşullar) : x^* noktası, $f: \mathbb{R}^n \rightarrow \mathbb{R}$ fonksiyonu $h(x) = 0$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$ kısıtı altında lokal minimum olsun .

Farz edelim ki $f, h \in C^2$ ve x^* regüler bir nokta olsun.

Bu durumda,

1) $Df(x^*) + \lambda^* Dh(x^*) = 0^T$

2) $\forall y \in T(x^*)$ için $y^T D^2l(x^*, \lambda^*) y \geq 0$ koşulu altında $\lambda^* \in \mathbb{R}^m$ vardır [10].

Teorem 2.1.2.3 (İkinci Mertebeden Yeter Koşul) : $f, h \in C^2$ ve $x^* \in \mathbb{R}^n$ olsun. $Df(x^*) + \lambda^* Dh(x^*) = 0^T$ ve $\forall y \in T(x^*)$, $y \neq 0$ için $y^T D^2l(x^*, \lambda^*) y > 0$ olacak şekilde $\lambda^* \in \mathbb{R}^m$ var ise x^* noktası $h(x) = 0$ kısıtı altında f fonksiyonunun kesin (strict) lokal minimumudur [10].

2.2 Eşitsizlik Kısıtlı Optimizasyon Problemleri

Eşitsizlik kısıtlı optimizasyon problemleri aşağıdaki gibi tanımlanır:

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & \\ & h(x) = 0 \\ & g(x) \geq 0 \end{aligned} \tag{2.2}$$

olmak üzere $f(x): \mathbb{R}^n \rightarrow \mathbb{R}$, $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$ ve $g(x): \mathbb{R}^n \rightarrow \mathbb{R}^p$ ele alalım.

Tanım 2.2.1: Bir $g_j(x) \geq 0$ eşitsizlik kısıtının x^* noktasında aktif kısıt olabilmesi için yeter şart $g_j(x^*) = 0$ olmasıdır. Tanım gereği $h(x^*) = 0$ eşitlik kısıtları her zaman aktiftir.

Tanım 2.2.2: x^* noktası $h(x) = 0$ ve $g(x) \geq 0$ kısıtlarını sağlayan bir nokta olmak üzere aktif eşitsizlik kısıtlarının indis kümesi $J(x^*)$ ile tanımlanır.

$$J(x^*) = \{j : g_j(x^*) = 0\}$$

biçiminde gösterilir. Eğer $\nabla h_i(x^*)$, $\nabla g_j(x^*)$, $1 \leq i \leq m$, $j \in J(x^*)$ vektörleri lineer bağımsız ise x^* noktası düzgün bir noktadır.

Şimdi lokal minimum için birinci mertebeden gerek şart verilecektir. Bu koşula Karush-Kuhn-Tucker (K.K.T) koşulu denir.

Teorem 2.2.3 (Karush-Kuhn-Tucker Teoremi) : $f, h, g \in C^1$, x^* düzgün bir nokta ve $g(x^*) \geq 0$ kısıtları altında f amaç fonksiyonunun lokal minimumu olsun. Bu durumda

1. $\mu^* \geq 0$
2. $Df(x^*) + \lambda^{*T} Dh(x^*) + \mu^{*T} Dg(x^*) = 0^T$
3. $\mu^{*T} g(x^*) = 0$

koşullarını sağlayan $\lambda^* \in \mathbb{R}^m$ ve $\mu^* \in \mathbb{R}^p$ vardır. Burada λ^* Lagrange çarpanı , μ^* K.K.T çarpanı denir [10].

Teoremin ifadesine dikkat edilirse $\mu_j \geq 0$ ve $g(x) \geq 0$ verilmiştir. Dolayısıyla (3) nolu koşulun $\mu_j^* = 0, g_j(x^*) = 0$ sağlanabilmesi için $g_j(x^*) > 0$ için $\mu_j^* = 0$ veya $g_j(x^*) = 0$ için μ_j^* 'nin negatif olmaması ($\mu_j^* > 0$) gerekir. Dolayısıyla eşitsizlik kısıtı aktif değilse $\mu_j^* = 0$ olmalıdır.

KKT koşullarını beş kısımda ele alabiliriz.

1. $\mu^* \geq 0$
2. $Df(x^*) + \lambda^{*T} Dh(x^*) + \mu^{*T} Dg(x^*) = 0^T$
3. $\mu^{*T} g(x^*) = 0$
4. $h(x^*) = 0$
5. $g(x^*) \geq 0$

İkinci Mertebeden Koşullar

İkinci mertebeden koşullardan önce bazı tanımları verelim. Lagrange fonksiyonunun Hesse Matrisi

$$L(x, \lambda, \mu) = F(x) + \lambda H(x) + \mu G(x)$$

ile tanımlayalım. Aktif kısıtların tanımlandığı tanjant uzayı ise

$$T(x^*) = \{y \in \mathbb{R}^n : Dh(x^*)y = 0, Dg_j(x^*)y = 0, j \in J(x^*)\}$$

ile tanımlanır [10].

Teorem 2.2.4 (İkinci Mertebeden Gerek Şartlar): x^* noktası $f: \mathbb{R}^n \rightarrow \mathbb{R}$ fonksiyonunun $h(x) = 0, g(x) \geq 0, h: \mathbb{R}^n \rightarrow \mathbb{R}^m, m \leq n, g: \mathbb{R}^n \rightarrow \mathbb{R}^p$ ve f koşulları altında lokal minimum olsun. Varsayalım ki x^* regülerdir. Bu durumda aşağıdaki koşulları sağlayacak $\lambda^* \in \mathbb{R}^m$ ve $\mu^* \in \mathbb{R}^p$ mevcuttur:

1. $\mu^* \geq 0, Df(x^*) + \lambda^{*T} Dh(x^*) + \mu^{*T} Dg(x^*) = 0^T, \mu^{*T} g(x^*) = 0$
2. $\forall y \in T(x^*)$ için $y^T L(x^*, \lambda^*, \mu^*) y \geq 0$ [10].

Eşitsizlik kısıtlarının ekstremum problemleri için ikinci mertebeden yeter koşulları belirlemek için kullanılacak kümeler aşağıda tanımlanmıştır [10].

$$\tilde{T}(x^*, \mu^*) = \{y : Dh(x^*)y = 0, Dg_i(x^*)y = 0, i \in \tilde{J}(x^*, \mu^*)\}$$

Öyle ki

$$\tilde{J}(x^*, \mu^*) = \{i : g_i(x^*) = 0, \mu_i^* > 0\} .$$

Dikkat edilirse $\tilde{J}(x^*, \mu^*) \subset J(x^*)$ şeklindedir. Bu da $T(x^*) \subset \tilde{T}(x^*, \mu^*)$ dır.

Teorem 2.2.5 (İkinci Mertebeden Yeter Koşullar): Farz edelim ki $f, h, g \in C^2$ ve $x^* \in \mathbb{R}^n$ uygun nokta, $\lambda^* \in \mathbb{R}^m$ ve $\mu^* \in \mathbb{R}^p$ vektörleri için

1. $\mu^* \geq 0, Df(x^*) + \lambda^{*T} Dh(x^*) + \mu^{*T} Dg(x^*) = 0^T, \mu^{*T} g(x^*) = 0$

2. $\forall y \in \tilde{T}(x^*, \mu^*), y \neq 0$ için $y^T L(x^*, \lambda^*, \mu^*) y > 0$

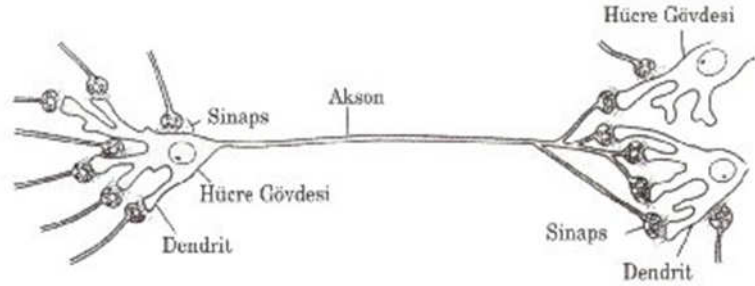
koşulları sağlansın. Bu durumda x^* noktası $h(x) = 0, g(x) \geq 0$ kısıtları altında f 'nin tam lokal minimumudur [11].

3. YAPAY SİNİR AĞLARI

İnsan beynindeki; öğrenme, öğrendiğini yeni durumlarda kullanma, hatırlama, karşılaştığı yeni durumlar için fikir yürütebilme gibi özellikleri bilgisayar ortamına taşınmasıyla oluşan bilgisayar sistemlerine yapay sinir ağları (YSA) denir. Söylediğimiz gibi yapay sinir ağları insan beynindeki sinir ağlarından esinlenilerek ortaya çıkmıştır.

3.1 YSA'nın Yapısı

Yapay sinir ağlarının yapısını daha iyi anlayabilmek için önce biyolojik sinir hücresinin (nöron) yapısını inceleyelim.



Şekil 3.1: Yapay Sinir Hücresi

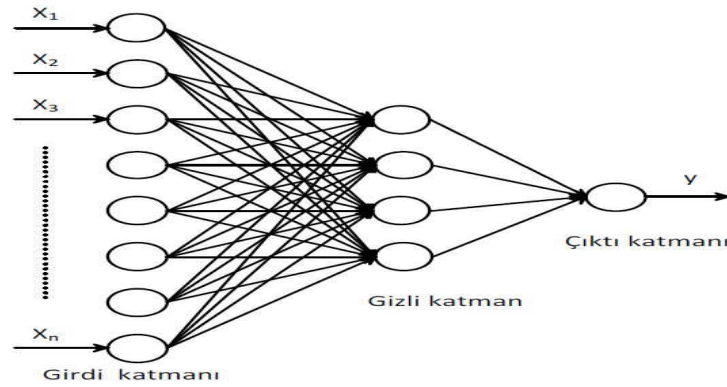
Şekil 3.1'de görüldüğü üzere bir sinir hücresi nöron gövdesi (soma), dendritler, akson ve presinaptik (sinaps öncesi) bağlantılar olmak üzere 4 kısımdan oluşur. Bu kısımda öncelikle değinmemiz gereken bir konuda sinaptır. Sinaps iki sinir hücresi arasında bilgi transferi sağlanırken oluşan sinirler arasındaki boşluktur. Bir sinir hücresi diğer sinir hücresinden gelen uyarıyı sinaps aracılığıyla dendritlerine alır. Dendritler uyarıyı somaya (yani nöron gövdesine) iletir. Soma uyarıları kuvvetlendirme ve zayıflatma etkilerine göre işler. Eğer uyarılar birbirini yeteri kadar kuvvetlendirerek belirli bir eşik değerini aşabilirse, aksona uyarı gönderilerek sinir

aktif hale geçirilir. Aksi durumda sinyal gönderilmez ve sinir pasif durumda kalır. Sinir hücreleri bir araya geldiklerinde bir ağ oluşturur. Bir sinir ağında milyonlarca nöron bulunabilir.

Yapay sinir ağları da biyolojik sinir ağlarının bilgisayara aktarılmış bir kopyasıdır. Nöronların belli bir doğrultuda bir araya gelmelerinden katmanlar oluşur. Katmanların farklı şekilde bağlanmalarıyla farklı ağ yapıları meydana gelir. Bu katmanlar sıra ile;

- Girdi katmanı
- Ara (gizli) katman
- Çıktı katmanı

olmak üzere üç bölümde incelenir.



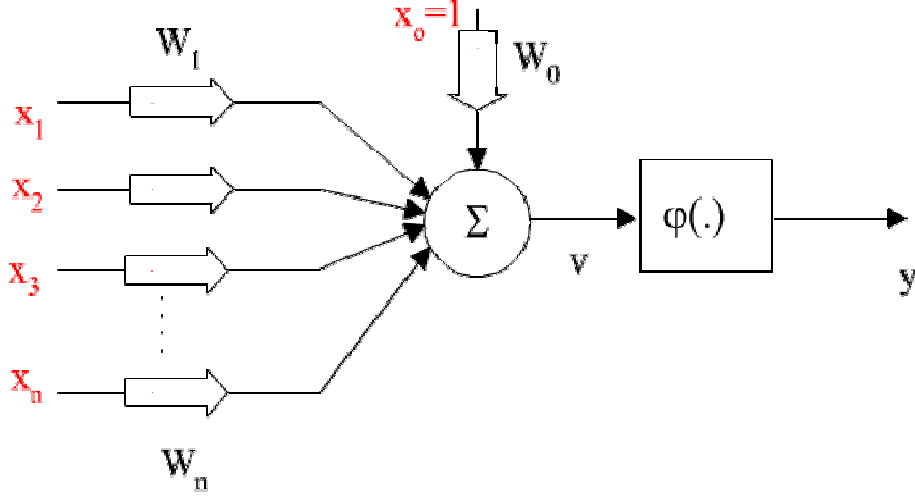
Şekil 3.2: Yapay Sinir Ağının Yapısı

Bu katmanlar ve her katman içinde paralel olarak bir araya gelerek ağı oluşturur.

- **Girdi katmanı:** Bu katmanlar dış dünyadan bilgileri alarak ara katmana iletmekle sorumludur.
- **Ara katmanlar:** Girdi katmanından gelen bilgiler işlenerek çıktı katmanına gönderilir. Bir ağ için birden fazla ara katman olabilir.
- **Çıkış katmanı:** Bu katman, ara katmandan gelen bilgileri işleyerek ağın girdi katmanından sunulan girdi seti için üretmesi gereken çıktıyı üretirler.

3.2 Yapay Sinir Hücresi (Proses Elemanı)

Biyolojik sinir ağlarında olduğu gibi yapay sinir ağlarının da sinir hücreleri vardır. Her sinir hücresinin (proses elemanının) 5 temel elemanı vardır.

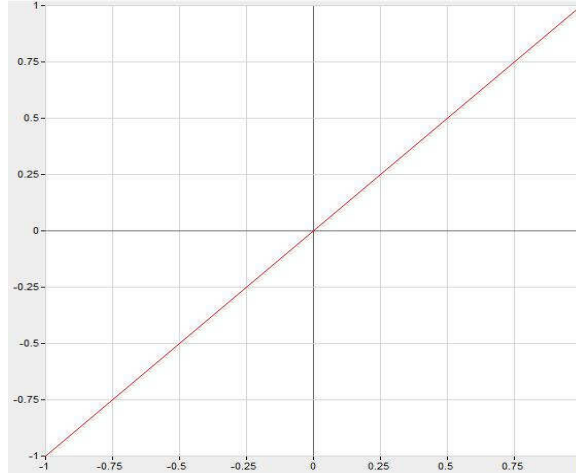


Şekil 3.3: Yapay Sinir Hücresinin Yapısı

Bunlar:

1. **Girdiler:** Bir yapay sinir hücresine dışarıdan verilen bilgilerdir. Bu bilgiler başta söylediğimiz gibi ağın öğrenmesini sağlayan bilgiler olacağı gibi diğer hücrelerden gelen bilgiler de olabilir.
2. **Ağırlıklar:** Yapay sinir hücresine gelen bilgiler girdilerin üretilecek çıktı üzerindeki etkisini ayarlayabilmek için çekirdeğe ulaşmadan önce geldikleri bağlantıların ağırlıklarıyla çarpılarak çekirdeğe iletilir.
3. **Toplama Fonksiyonu (Birleştirme Fonksiyonu):** Bir yapay sinir hücresine ağırlıklarla çarpılarak gelen girdileri toplayarak o hücrenin net girdisini hesaplayan fonksiyondur.
4. **Aktivasyon Fonksiyonu:** Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşı üreteceği çıktıyı belirler. Aktivasyon fonksiyonu genellikle doğrusal olmayan bir fonksiyon seçilir. Aşağıda bazı aktivasyon fonksiyonları verilmiştir:

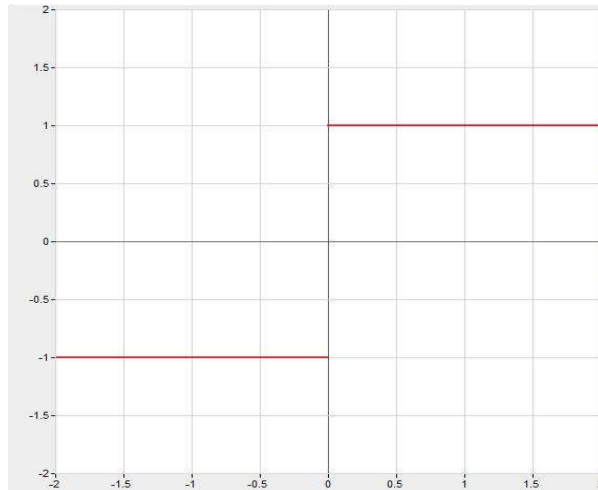
Doğrusal Aktivasyon Fonksiyonu: Doğrusal fonksiyonlar $F(Net) = A * Net$ biçiminde genellenebilir. A sabit bir katsayıdır. Toplama fonksiyonundan çıkan sonuç, belli bir katsayı ile çarpılarak hücrenin çıktısı olarak hesaplanır.



Şekil 3.4: Doğrusal Aktivasyon Fonksiyonu Grafiği

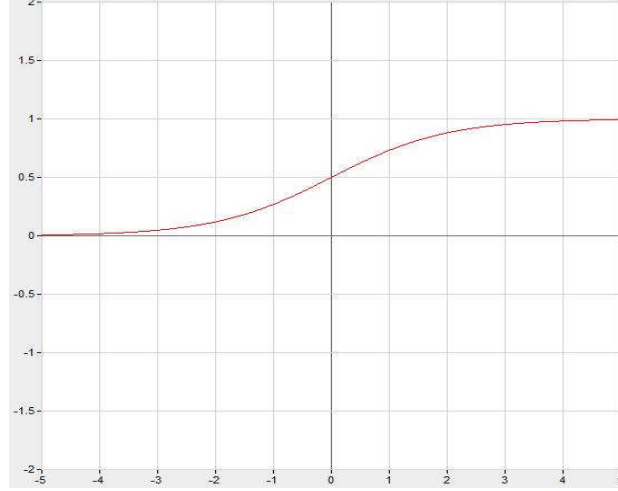
Adım (Step) Aktivasyon Fonksiyonu: $F(Net) = \begin{cases} 1 & \text{if } Net > \text{Eşik Değer} \\ 0 & \text{if } Net \leq \text{Eşik Değer} \end{cases}$

Gelen Net girdinin belirlenen bir eşik değerinin altında veya üstünde olmasına göre hücrenin çıktısı 1 veya 0 değerini alır.



Şekil 3.5: Adım Aktivasyon Fonksiyonu Grafiği

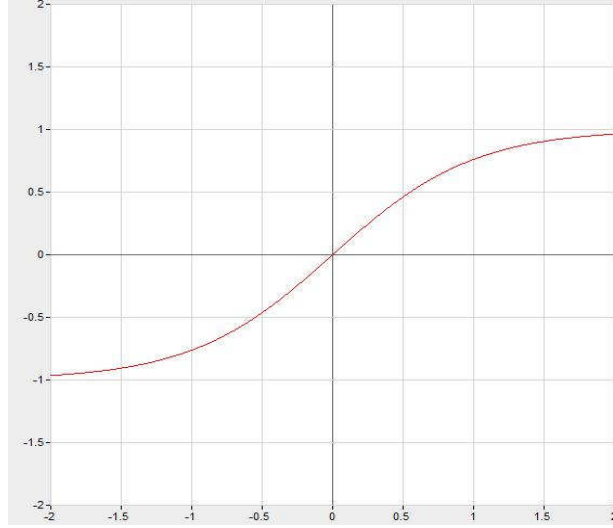
Sigmoid Aktivasyon Fonksiyonu: $F(NEt) = \frac{1}{1 + e^{-Net}}$ Sigmoid aktivasyon fonksiyonu sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmayışı nedeniyle yapay sinir ağı uygulamalarında en sık kullanılan fonksiyondur. Bu fonksiyon girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir.



Şekil 3.6: Sigmoid Aktivasyon Fonksiyonu Grafiği

Tanjant Hiperbolik Aktivasyon Fonksiyonu: $F(NEt) = \frac{e^{Net} + e^{-Net}}{e^{Net} - e^{-Net}}$

Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.



Şekil 3.7: Tanjant Hiperbolik Aktivasyon Grafiği

Aktivasyon fonksiyonu seçilirken dikkat edilmesi gereken bir diğer nokta ise fonksiyonun türevinin kolay hesaplanabilir olmasıdır. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi de kullanıldığı için hesaplamaların yavaşlamaması için türevi kolay hesaplanabilir bir fonksiyon seçilir. Günümüzde en yaygın olarak kullanılan “Çok katmanlı algılayıcı” modelinde genel olarak aktivasyon fonksiyonu olarak “Sigmoid fonksiyonu” kullanılır.

5. **Hücrenin Çıktısı:** Aktivasyon fonksiyonundan çıkan değer hücrenin çıktı değeri olmaktadır. Bu değer ister yapay sinir ağının çıktısı olarak dış dünyaya verilsin isterse tekrardan ağın içinde kullanılabilir. Her hücrenin birden fazla girdisi olmasına rağmen bir tek çıktısı olmaktadır. Bu çıktı istenilen sayıda hücreye bağlanabilir.

3.2.1 Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağlarının özellikleri uygulanan ağa göre değişiklik göstermektedir. Burada yapay sinir ağlarının tümünde geçerli olan genel özellikler verilecektir. Bunları aşağıdaki gibi sıralayabiliriz:

- **Yapay sinir ağları makine öğrenmesi gerçekleştirirler:** Yapay sinir ağları olayları öğrenerek benzer olaylar karşısında benzer kararlar vermeye çalışırlar.

- **Programları çalışma sitili bilinen programlama yöntemlerine benzememektedir.** Geleneksel programlama ve yapay zeka yöntemlerinden farklı bir bilgi işleme yöntemleri vardır.
- **Bilginin saklanması:** Diğer programlarda olduğu gibi veriler bir veri tabanında veya programın içinde gömülü değil, ağın üzerinde saklı olup ortaya çıkartılması ve yorumlanması zordur.
- **Yapay sinir ağları örnekleri kullanarak öğrenirler.** Yapay sinir ağları olaylar hakkında genelleme yapabilmesi için öncelikle o olayla ilgili örnekler kullanılarak ağ genelleme yapabilecek yeteneğe kavuşturulmalıdır. Elde edilen örneklerin olayın tamamı ile gösterebilmesi çok önemlidir. Ağa olay bütün yönleriyle gösterilmez ve ilgili örnekler sunulmazsa başarılı sonuçlar elde edilemez.
- **Yapay sinir ağlarının güvenle çalıştırılabilmesi için önce eğitilmeleri ve performanslarının test edilmesi gerekmektedir.** Ağı eğitmek için elde edilen örnekler iki ayrı sete bölünerek, ilki ağı eğitmek için diğeri de ağın performansını ölçmek için kullanılır.
- **Görülmemiş örnekler hakkında bilgi üretebilirler.** Ağ kendisine gösterilen örneklerden genelleme yaparak görmediği örnekler için bilgiler üretebilir.
- **Algılamaya yönelik olaylarda kullanılabilir.** Ağlar daha çok algılamaya yönelik bilgileri işlemede kullanılır.
- **Şekil ilişkilendirme ve sınıflama yapabilirler.**
- **Örüntü tamamlama gerçekleştirebilirler.** Bazı durumlarda ağa eksik bilgi veya şekil verilip, eksik bilgileri bulması istenir.
- **Kendi kendini organize etme ve öğrenebilme yetenekleri vardır.**
- **Eksik bilgi ile çalışabilmektedirler.** Yapay sinir ağları eksik bilgiyle de çalışmaya devam ederler. Fakat burada eksik bilginin önemine bakılmalıdır, bunu da kullanıcı bilemez. Eksik bilginin önemli olup olmadığına ağın performansına bakılarak karar verilir. Eğer ağın performansı değişmiyorsa, eksik bilgi önemli değildir. Aksi söz konusu ise eksik bilgi önemlidir.
- **Hata toleransına sahiptir.** Yapay sinir ağlarının eksik bilgiyle çalışması hatalara karşı toleranslı olmalarını sağlamaktadır. Ağın bazı hücrelerinin bozulması ve çalışmaması halinde ağ çalışmaya devam eder. Yine burada bozuk olan hücrenin önemini anlamak için, ağın performansına dikkat etmek gerekir.
- **Belirsiz, tam olmayan bilgileri işleyebilmektedirler:** Yapay sinir ağlarının belirsiz bilgileri işleyebilme yetenekleri vardır.
- **Dereceli bozulma gösterirler:** Yapay sinir ağlarının hatalara karşı toleranslı olması bozulmalarının da dereceli olmasına sebep olmaktadır.
- **Dağıttık belleğe sahiptirler:** Yapay sinir ağlarında bilgi ağa yayılmış durumdadır. Hücrelerin birbirleri ile bağlantılarının değerleri ağın bilgisini göstermektedir.
- **Sadece nümerik bilgiler ile çalışabilmektedirler.** Sembolik ifadeler ile gösterilen bilgilerin nümerik gösterime çevrilmeleri gerekmektedir [8].

3.2.2 YSA'ların Tarihçesi

Burada yapay sinir ağı hakkında belli başlı önemli çalışmalar ve tarihlerine yer verilmiştir [8].

- İlk yapay sinir ağı modeli 1943 yılında, bir sinir hekimi olan Warren McCulloch ile bir matematikçi olan Walter Pitts tarafından gerçekleştirilmiştir [12].
- 1949 yılında Hebb "Organization of Behavior" adlı eserinde öğrenme ile ilgili temel teoriyi ele almıştır .
- 1957 yılında Frank Rosenblatt'ın [13] Perceptron'u geliştirmiştir.
- 1959 yılında Bernard Widrow ve Marcian Hoff (Stanford Üniversitesi) ADALINE (Adaptive Linear Neuron) modelini geliştirmişler ve bu model YSA'ların mühendislik uygulamaları için başlangıç kabul edilmiştir [14].
- 1960'ların sonlarına doğru Grosberg Carpenter ile birlikte Adaptif Rezonans Teorisini (ART) geliştirmiştir [15,16].
- 1970'lerin sonlarına doğru Fukushima, NEOCOGNITRON modelini tanıtmıştır [17,18].
- 1982-1984 yıllarında Hopfield tarafından yayınlanan çalışmalar ile YSA'ların genelleştirilebileceği ve çözümü zor problemlere çözüm üretebileceğini göstermiştir [19,20].
- 1988 yılında, Broomhead ve Lowe radyal tabanlı fonksiyonlar modelini (Radial Basis Functions RBF) geliştirmişler ve özellikle filtreleme konusunda başarılı sonuçlar elde etmişlerdir [21].

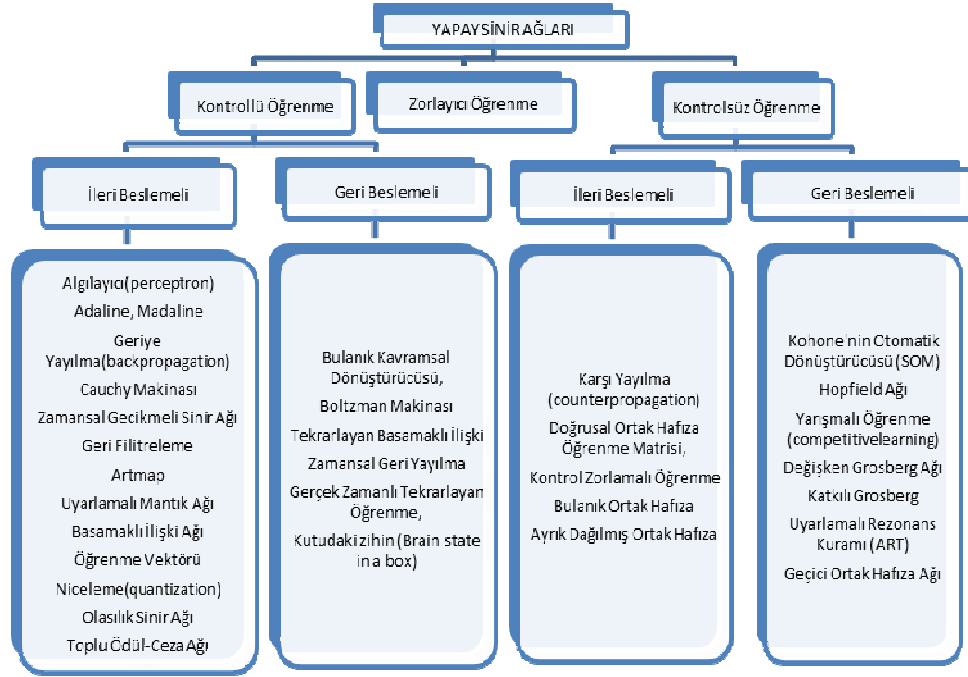
3.2.3 YSA'ların Kullanıldığı Alanlar

Başarılı uygulamalar incelendiğinde, YSA 'ların çok boyutlu, gürültülü, karmaşık, kesin olmayan, eksik, kusurlu, hata olasılığı yüksek sensör verilerinin olması ve problemi çözmek için matematiksel modelin ve algoritmaların bulunmadığı, sadece örneklerin var olduğu durumlarda yaygın olarak kullanıldıkları görülmektedir. Bu amaçla geliştirilmiş ağlar genellikle şu fonksiyonları gerçekleştirmektedirler;

- Muhtemel fonksiyon kestirimleri
- Sınıflandırma
- İlişkilendirme veya örüntü eşleştirme
- Zaman serileri analizleri
- Sinyal filtreleme
- Veri sıkıştırma

- Örüntü tanıma
- Doğrusal olmayan sinyal işleme
- Doğrusal olmayan sistem modelleme
- Optimizasyon
- Kontrol [8].

YSA'ların birçok kullanım alanı vardır. Bunlar kısaca Şekil 3.8'de gösterilmiştir.



Şekil 3.8: Yapay Sinir Ağı Türleri [22]

Genellikle optimizasyon problemlerinin iki çeşidi sinir ağları kullanılarak çözülebilir. Bunlardan ilki mobil robotların kontrolüdür, buradaki sinir ağı algısal girdiler ve davranışları arasındaki ilişkiyi öğrenmede kullanılır [23]. İkinci çeşidi ise klasik optimizasyondur, buradaki sinir ağı bir enerjinin ya da amaç fonksiyonunu minimize eden bir denge noktası bulmakta kullanılır [24]. Oluşturulan dinamik sistem yapısal optimizasyon için sinir dinamik modelini tanımlayan diferansiyel denklem sistemidir. Bu aslında yapay sinir ağları modelinin öğrenme kuralıdır [24]. Oluşturulan dinamik sistemin kararlılığı Lyapunov fonksiyonu ile gösterilir.

Sinir ağları örneklerden “öğrenir” ve örnekleme verileri (eğitim seti) üzerinde “genelleştirme” yeteneğini kullanır. Genelleştirme, sinir ağının örnekleme veri seti içinde yer almayan yeni verileri enterpole yada ekstrapole edebilme yeteneğidir. Bir sinir ağının “gücü”, örnekleme verilerinden ne kadar iyi genelleştirme yapabildiğine

bağlıdır. Öğrenme algoritması; istenen (hedef) çıkış vektörü ile gerçekleşen çıkış vektörü arasındaki fark cinsinden bir ölçü değeri olan hata fonksiyonunu kullanarak; deneme (eğitim) seti üzerindeki ortalama hatayı azaltacak şekilde, ağ içindeki ağırlıkları dengelemektedir. Bu doğru olarak gerçekleştirildiğinde sinir ağı, yeni giriş verileri için istenen sonuçları doğru olarak tahmin edecektir [25].

4. OPTİMİZASYON VE YAPAY SİNİR AĞLARI

Son yıllarda, yapay sinir ağları matematiksel programlama ya da optimizasyon problemlerini çözmek için oldukça geliştirilmiştir. Sinir ağlarından bazıları optimizasyon için kullanılmıştır, örneğin Hopfield sinir ağları [19], kendini örgütleyen özellik haritaları (self-organizing feature maps), ve Boltzmann makineleri gibi [26].

Hopfield [19] sinir ağının kararlılığı için bir Lyapunov enerji fonksiyonu kullanmıştır [26]. Bu tek katmanlı Hopfield ağı yalnız iki terimli girdiler üzerinde ve farklı zamanlardaki çıktı kalıplarında kullanılabilir. Cohen ve Grossberg ve Hopfield sürekli değişkenler ve cevaplar üzerindeki çalışmaları sürekli Hopfield ağı olarak geliştirdiler. Fakat bu sürekli Hopfield ağının bir limite sahip olması için bağlantı ağırlığının simetrik olması gerekmektedir. Hopfield ve Tank [27] düğümün aktifleştirilmesini tanımlayan diferansiyel denklemlerin kullanımında enerji fonksiyonunu minimize etmek için bir ağ önermişlerdir. Önermiş oldukları bu ağın donanım uygulaması için sürekli değişken birimler kullanan elektrik devrelerinden yararlanmışlardır. Bu modelde lokal minimum için Kunh-Tuker gereklilik koşulları sağlanmamaktadır. Bu da uygulamada optimizasyon şartlarını sağlamayan geçersiz çözümlere yaklaşmasına sebep olmuştur [26].

Sonrasında Hopfield sinir ağları ve yapay sinir ağları optimizasyon problemlerinin hemen hemen bütün sınıfları için geliştirilmiştir. Hopfield yapay sinir ağının optimizasyon yaklaşımı enerji fonksiyonunun dinamik sistemini ele alıp, ağların davranışlarının özelliklerini saptayıp çözülmesi gereken problemleri temsil eder. Hopfield ağlarının genişletilmesi ile yapay sinir ağlarının ilgili alanlarının da yenilenmesi nedeniyle çeşitli sinir ağları önerilmiştir. Kennedy ve Chua [28] Tank ve Hopfield'ın doğrusal olmayan problem çalışmalarını genişletmişlerdir. Rodriguez-Vazquez ve arkadaşları [29,30] doğrusal ve doğrusal olmayan programlama problemlerinin çözümü için anahtarlama-kapasitör (switched-capacitor) ağlarını sunmuşlardır. Daha sonra Zhang ve Constantinides [31] genel doğrusal olmayan programlama problemlerinin çözümü için Lagrange sinir ağlarını tanıtmıştır, Cichocki ve Unbehauen [32] doğrusal olmayan programlama problemi çözümünde

Lagrange çarpanı tabanlı sinir ağını sunmuşlardır. Xia ve arkadaşları [33] lineer ve kuadratik programlama problemlerinin çözümü için primal ve dual metodu önermişlerdir. Şimdi optimizasyon problemlerin çözümü için bazı metotları inceleyeceğiz.

4.1 Ceza (Penalty) Fonksiyonu Metodları

Ceza fonksiyonu metodu kısıtlı yada kısıtsız optimizasyon problemleri için kullanılabilen oldukça ilgi çeken bir yöntemdir. Ceza metodu dik iniş yaklaşım yönünü kullanan bir yöntemdir. Şimdi doğrusal ve doğrusal olmayan programlama problemlerini çözümlerinde ceza metodlarının YSA'larda uygulamasını inceleyeceğiz.

4.1.1 Doğrusal Program (LP) Problemleri

Genel doğrusal programlama (LP) problemlerinin primal ve dual LP formu aşağıdaki gibi gösterilir [26];

$$\begin{aligned} \min f(x) &= c^T x \\ \text{s.t.} \\ Ax &\geq b (= g(x) \geq b_j, j = 1, 2, \dots, m) \\ \text{ve } x &\geq 0 \\ x, c &\in \mathbb{R}^{n \times 1}, A \in \mathbb{R}^{m \times n}, \text{ ve } b \in \mathbb{R}^{m \times 1}; \end{aligned}$$

$$\begin{aligned} \max g(y) &= b^T y, \\ \text{s.t. } A^T y &\leq c (= f_i(y) \leq c_i, i = 1, 2, \dots, n) \\ \text{ve } y &\geq 0, \\ c \in \mathbb{R}^{n \times 1}, A &\in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m \times 1}, \text{ ve } y \in \mathbb{R}^{m \times 1}. \end{aligned}$$

İlk olarak doğrusal program problem yapısını Hopfield ve Tank [34] sunmuşlardır ve enerji fonksiyonunu aşağıdaki gibi tanımlamışlardır [26]:

$$E_{TH} = f(x) + \sum_{j=1}^m (g_j^+(x))^2 + \sum_{i=1}^n x_i^2 / 2sR_{ii}$$

R bir $n \times n$ diyagonal bir matris, $s > 0$ penalty parametresi ve $g^+ = [g_1^+, g_2^+, \dots, g_m^+]$, $g_j(x) < b_j$ için bir penalty vektörüdür. s penalty parametresinin yeterince büyük olması gerekir.

Kennedy ve Chua [35] Tank ve Hopfield'in sinir ağını geliştirerek, hatalı (inexact) ceza fonksiyonunu içeren bir sinir ağı modelini önerdiler. Bu modelin enerji fonksiyonu;

$$E_{KC} = f(x) + \frac{s}{2} \sum_{j=1}^m (g_j^+(x))^2$$

$s > 0$ ceza parametresidir.

Rodriguez-Vazquez ve arkadaşları [29] kısıtsız bir optimizasyon problemini doğrusal program problemine dönüştürmek için farklı bir penalty metodu kullandılar. Bu modelin enerji fonksiyonu;

$$E_{RV}(x, \alpha) = f(x) + \left| \sum_{j=1}^m \min \{0, g_j(x) - b_j\} \right|$$

$\alpha > 0$ dır. Bu ifadeyi sağlayan fonksiyon negatif olmamalıdır. Her bir k adımı için $x_i(k) = \max \{x_i(k), 0\}$ dır. Uygun bölgeye ulaşıldığında eğri amaç fonksiyonunun minimumuna doğru gider.

Maa ve Shanblatt [36] problemlerin çözümü için iki aşamalı sinir ağı yapısını kullanmışlardır. İlk aşamada $t < t_1$, t_1 rastgele seçilir ve ağın yapısı Kennedy ve Chua modelinin aynısıdır. Ağın kararlılığı t_1 zaman parametresinin ve s ceza parametrelerinin nasıl seçildiğine bağlıdır. Chong ve arkadaşları tam diferansiyellenemeyen ceza fonksiyonlarına dayanan dinamik gradyant yaklaşımlarını kullanarak doğrusal programlama problemlerinin çözümü için sinir ağları modellerinin bir sınıfını analiz etmişlerdir.

4.1.2 Doğrusal Olmayan Programlama (NLP) Modelleri

Doğrusal olmayan programlama (NLP) problemlerinin gerçek hayatta bir çok uygulamaları vardır ve hem teorik hem de pratik yönüyle oldukça dikkat çekmektedir. NLP problemlerini hesaplamak oldukça karmaşıktır, sinir ağı ise daha hızlı bir yaklaşım sunan ceza fonksiyonu ile yaklaşır [26]. Genel bir NLP yapısı aşağıdaki gibi gösterilebilir [37];

$$\begin{aligned} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i=1,2,\dots,l \quad \text{koşulu ile} \\ & g_j(x) \leq 0 \quad j = 1,2,\dots,m \end{aligned}$$

$f, g_1, g_2, \dots, g_m, h_1, h_2, \dots, h_l$ \mathbb{R}^n üzerinde tanımlı fonksiyonlardır. Genellikle $f_p(x)$ penalty fonksiyonu [26];

$$f_p(x) = f(x) + \sum_{j=1}^m \kappa_j \zeta_j [g_j(x)] + \sum_{i=1}^l \kappa_i \zeta_i [h_i(x)]$$

biçiminde gösterilir. κ_j ve κ_i penalty parametreleri, ζ_j ve ζ_i penalty terimleridir.

Effati ve Beymain [38] NLP problemlerinin kompleks çözümü için daha içgüdüsel ve daha basit bir devirli sinir ağı sundular.

Günümüzde ceza metodları iç ve dış ceza metodları olarak incelenmektedir. Bunlara kısaca değinecek olursak; dış ceza metodları YSA'lar tarafından uygulamalarda basitlik sunduğundan daha çok tercih edilmektedir. Ayrıca minimizasyonun başlamasında uygun bir noktaya gerek duymaz, eşitlik ve eşitsizlik

kısıtlarının her ikisi içinde kolay kullanılabilirken iç ceza fonksiyonu eşitlik kısıtlarını içermez. Buna karşılık iç ceza fonksiyonu metodları kısıtların her zaman sağlanması mecburi olan problemler içinde kullanmak için elverişlidir.

Aşağıdaki tabloda şimdiye kadar penalty metodlarının çalışmaları hakkında kısa bilgi verilmiştir [26].

Tablo 4.1: Optimizasyon Problemleri için Ceza Fonksiyonu Metodları

Önerilen Metod	Problem Tipi	Aktivasyon Fonksiyonu	Penalty	Öğrenme Oranı	Başlangıç Durumu
Tank ve Hopfield (1986)	LP	Sigmoid Fonksiyonu	s yeteri kadar büyük olmalı	C pozitif diagonal matristir.	Mevcut değil
Kennedy ve Chua (1988)	LP ve NLP	Sigmoid Fonksiyonu	$s > 0, s \rightarrow \infty$	C pozitif diagonal matristir.	Mevcut değil
Rodríguez-Vázquez ve ark. (1988,1990)	LP ve NLP	$x_i(k+1) = \max\{x_i(k), 0\}$	$\alpha > 0, S_i = \begin{cases} 1, & g_i(x) > 0 \\ 0, & g_i(x) \leq 0 \end{cases}$	$0 < \mu_i < 1$	Herhangi bir başlangıç durumu
Maa ve Shablatt(1992)	LP ve NLP	Sigmoid Fonksiyonu	ϵ küçük bir pozitif değerdir ve $s > 0$	Mevcut değil	Uygun değişkenler
Chong ve arkadaşları (1999)	LP	Sigmoid Fonksiyonu	Seçim kuralına göre seçilmiş parametreler	Mevcut değil	Uygun değişkenler
Silva ve arkadaşları (2005)	NLP	Sigmoid Fonksiyonu	Mevcut değil	Mevcut değil	Uygun değişkenler
Effati ve Baymain (2005)	NLP	Mevcut Değil	Kısıtlı diferansiyel eşitliklerden seçilen	Mevcut değil	Uygun değişkenler

4.2 Lagrange Çarpanları Metodu

Bu metod penalty metodlarına benzemektedir. Lagrange çarpanları ve genişletilmiş Lagrange metodları hedef ağın amaç fonksiyonu ve kısıtların birleşiminden oluşmaktadır. Şimdi sırasıyla Doğrusal Programlama (LP) ve Doğrusal Olmayan Programa (NLP) problemlerinin çözümü için bu metodlara bakılacaktır.

4.2.1 Doğrusal Programlama (LP) Metodu

Zang ve Constantinides [31] Hopfield sinir ağının vasıtasıyla LP problemlerinin çözümü için Lagrange ve Genişletilmiş Lagrange metodlarını sunmuşlardır. Bu metodların enerji fonksiyonu;

$$E_L(x, \lambda) = f(x) + \sum_{j=1}^m \lambda(g_j(x))$$

ve

$$E_{aL}(x, \lambda, \kappa) = f(x) + \sum_{j=1}^m \lambda (g_j(x)) + \frac{1}{2} \sum_{j=1}^m \kappa |g_j(x)|^2,$$

sırasıyla λ ve κ , Lagrange çarpanları vektörü ve penalty parametreleridir.

Gill ve arkadaşları [39] eğer fonksiyon konveks ise gradyant temelli araştırma yaklaşımlarının lokal bir minimuma yakınsadığının garanti olabileceğini belirttiler. Zhang ve Constantinides [31] κ ceza parametresinin yakınsaklık için 5'ten daha büyük olmaması gerekmektedir. Shih ve arkadaşları [40] LP yaklaşımına dayanan çok amaçlı ve çok katmanlı problemlerin çözümü için genişletilmiş (augmented) Lagrange çarpanları metodunu tanıttılar. Bu metodun enerji fonksiyonu,

$$E_S(x, \lambda) = c^T x + \lambda^T (Ax - b) - \frac{1}{2} \lambda^T \lambda + \frac{\kappa}{2} (Ax - b)^T (Ax - b)$$

fonksiyon ceza parametrelerini, Lagrange çarpanını ve düzenlenmiş terimleri içerir.

4.2.2 Doğrusal Olmayan Program (NLP) Problemleri

Lagrange çarpanları metodları NLP problemlerinin çözümlerinde yaygın olarak kullanılmaktadır. Sinir ağı yaklaşımı enerji fonksiyonunu elde etmek için

Lagrange çarpanını $f_L(x, \lambda) = f(x) + \sum_{i=1}^l \lambda h_i(x)$, $x \in R^{n \times 1}$ ve $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_l]^T$

kullanır.

Çözüm prosedürü dik iniş yaklaşımı yoluyla eşitliklerin dinamik sistemini kullanır[40]. Eğer amaç fonksiyonu konveks fonksiyon değilse bu prosedür kolayca lokal minimum bölgesinde kısılı kalır. Ham ve Kostanic [41] bir α gürültü faktörünün $\lambda(k+1) = \lambda(k) + \mu_\lambda \left[\partial f_L(x(k), \lambda(k)) / \partial \lambda(k) - \alpha \lambda(k) \right]$, $0 \leq \alpha \leq 1$ olarak dalga salınımını azaltabileceğini belirtmişlerdir.

Gong ve arkadaşları [42] konveks optimizasyon problemlerinin çözümü için bir değiştirilmiş (modified) Lagrange metodu sunmuşlardır. Sinir ağının dinamiği orijinal problemin Karush-Kuhn-Tucker (KKT) koşullarının sağlandığı denge

noktasını gösterir. Wu ve Tam [43] Lagrange çarpanına dayanan kuadratik programlama problemlerinin çözümü için değişik bir sınır ağı modeli sunmuşlardır.

Diğer forma ekstra ceza terimleri eklenerek genişletilmiş Lagrange çarpanları metodu elde edilmiştir. Elde edilen bu form

$$f_{aL}(x, \lambda, \kappa) = f(x) + \sum_{i=1}^l \lambda h_i(x) + \sum_{i=1}^l \kappa (h_i(x))^2,$$

λ ve κ sırasıyla Lagrange çarpanlarının ve penalty parametrelerinin vektörleridir [41]. Genişletilmiş Lagrange çarpanı eşitsizlik kısıtlı NLP problemleri için aşağıdaki biçimde büyütülebilir;

$$f_{aL}(x, \lambda, \kappa) = f(x) + \sum_{j=1}^m \lambda \max\{0, g_j(x)\} + \sum_{j=1}^m \frac{\kappa}{2} \max\{0, g_j(x)\}^2$$

$\lambda \in \mathfrak{R}^{m \times 1}$ ve $\kappa \in \mathfrak{R}^{m \times 1}$ sırasıyla Lagrange çarpanları ve penalty parametrelerinin vektörleridir. Kompakt formu aşağıdaki gibi ifade edilebilir [41]:

$$f_{aL}(x, \lambda, \kappa) = f(x) + \sum_{j=1}^m S_j \left[\lambda g_j(x) + \kappa/2 (g_j(x))^2 \right].$$

Eğer $g_j(x) > 0$ ise $S_j = 1$, diğer durumda $S_j = 0$ dir. Bu eşitlik Radriguez-Vazquez ve arkadaşlarının [29] önerdiği yapıya yakındır ve bu sistem bir optimal çözüm bulabilir.

Şimdi aşağıdaki tabloda kısaca bu metodlar hakkındaki çalışmalar verilecektir [26]:

Tablo 4.2: Optimizasyon Problemleri için Lagrange Çarpanları Metodları

Önerilen Metod	Problem Tipi	Aktivasyon Fonksiyonu	Penalty	Öğrenme Oranı	Başlangıç Durumu
Zhang ve Constantinides (1992)	LP ve NLP	Mevcut Değil	$0 < c < 5$	Mevcut değil	Herhangi bir başlangıç durumu
Gong ve ark. (1997)	NLP	Mevcut Değil	Mevcut değil	$\rho(t) = \frac{\rho_0}{1 + \alpha t}$ or $\rho(t) = \rho_0 \exp^{-\alpha t}$ ρ_0 büyük, α pozitifdir.	Genellikle başlangıç noktası için orijin kullanılır
Wu ve Tam (1999)	NLP	Sigmoid Fonksiyonu	$k > 0$	Mevcut değil	Herhangi bir başlangıç durumu

5. DEĞİŞTİRİLMİŞ BARIYER VE GENİŞLETİLMİŞ LAGRANGE METODU İLE GRADYANT TABANLI YAKLAŞIM

Değiştirilmiş (modified) bariyer ve genişletilmiş (augmented) Lagrange (DBGL) metodu Polyak'ın [44] değiştirilmiş bariyer fonksiyonu metodunun ve Hestenes'in [45] ve Powell'ın [46] eşitlik kısıtları için genişletilmiş Lagrange metodunun bir birleşimidir. Yani DBGL bir değiştirilmiş bariyer terimlerinin eşitsizlik kısıtlarını ve genişletilmiş Lagrange'ın eşitlik kısıtlarını içerir. DBGL fonksiyonu DBGL metodu içindeki bariyer ve ceza fonksiyonlarının en iyi özelliklerini korurken temel dezavantajlarını ortadan kaldırarak kullanır. İkinci mertebeden optimallik koşulları altında DBGL fonksiyonu bariyer ve ceza parametreleri yeterli büyüklüğe geldiğinde eşitlik kısıtlarının Lagrange çarpanlarının ve eşitsizlik kısıtlarının pozitif Lagrange çarpanlarının herhangi bir vektörü için tek bir global minimuma sahiptir. Her ne kadar genişletilmiş Lagrange orijinal olarak eşitlik kısıtlı problemler için tasarlanmış olsa da Rockafellar [47] tarafından eşitsizlik kısıtları içinde ele alınarak incelenmiştir. Amaç ve kısıt fonksiyonları yüksek diferansiyellenebilirliğe sahipken genişletilmiş Lagrange yalnız bir kere diferansiyellenebilir [48].

5.1 Problemin İfadesi ve Temel Varsayımları

Burada aşağıdaki problemi kullanacağız;

$$\begin{aligned} \min \quad & f_o(x) \\ \text{s.t} \quad & f_i(x) \geq 0 \quad i = 1, \dots, p \text{ koşulu ile,} \\ & g_j(x) = 0 \quad j = 1, \dots, q \end{aligned} \quad (5.1)$$

$x \in \mathbb{R}^n$ ve varsayalım ki f_o, f_1, \dots, f_p ve g_1, g_2, \dots, g_q $\mathbb{R}^n \rightarrow \mathbb{R}$ C^2 fonksiyonlarıdır.

Bu problem için klasik Lagrangian

$$L(x, u, v) = f_0(x) - u^T f(x) - v^T g(x),$$

$u \in \mathbb{R}_+^p$, $v \in \mathbb{R}^q$ ve $f(x) = (f_1(x), \dots, f_p(x))$ ve $g(x) = (g_1(x), \dots, g_q(x))$ kolon vektörleridir. \mathbb{R}_+^p \mathbb{R}^p 'nin negatif olmayan bölgesi anlamına gelir ve \mathbb{R}_{++}^p bu bölgeye dahildir.

x^* , (5.1) probleminin tam yerel ve $I^* = \{i : f_i(x) = 0\} = \{1, \dots, r\}$ bu noktadaki aktif eşitsizlik kısıtlarının indeks kümesi olsun.

İkinci mertebeden yeterlilik koşulları ;

- C1. $\nabla f_i(x^*)$, $i = 1, \dots, r$ ve $\nabla g_j(x^*)$, $j = 1, \dots, q$ gradyanları lineer bağımsızdır ve bir tek $w^* = (u^*, v^*) \in \mathbb{R}_+^p \times \mathbb{R}^q$ Lagrange çarpanı mevcuttur;

$$\nabla_x L(x^*, u^*, v^*) = \nabla f_0(x) - \sum_{i=1}^p u_i^* \nabla f_i(x^*) - \sum_{j=1}^q v_j^* \nabla g_j(x^*) = 0 \quad (5.2)$$

- C2. (x^*, u^*, v^*) 'da x 'e göre $L(x, u, v)$ Lagrangian'ın Hessian'i

$$\nabla_{xx}^2 L(x^*, u^*, v^*) = \nabla^2 f_0(x) - \sum_{i=1}^p u_i^* \nabla^2 f_i(x^*) - \sum_{j=1}^q v_j^* \nabla^2 g_j(x^*),$$

x^* 'da uygun kümede afin alt uzay teğet üzerinde pozitif tanımlıdır.

$$\begin{aligned} y^T \nabla_{xx}^2 L(x^*, u^*, v^*) y &> 0 \\ \forall y \in Y \subset \mathbb{R}^n \end{aligned} \quad (5.3)$$

$$Y = \{y : y^T \nabla f_i(x^*) = 0, i = 1, \dots, r, y^T \nabla g_j(x^*) = 0, j = 1, \dots, q\},$$

- C3. Güçlü tamamlayıcı yapay değişken eşitsizlik kısıtları içinde sağlanır; yani

$$u_i^* f_i(x^*) = 0, \quad i = 1, \dots, p \quad (5.4)$$

$$u_i^* > 0, \quad i = 1, \dots, r; \quad f_i(x^*) > 0, \quad i = r + 1, \dots, p. \quad (5.5)$$

Şimdi değiştirilmiş bariyer ve genişletilmiş Lagrange fonksiyonunu tanımlayacağız [48].

5.2 Değiştirilmiş Bariyer ve Genişletilmiş Lagrange Fonksiyonu

Değiştirilmiş bariyer ve genişletilmiş Lagrange fonksiyonu $F(x, u, v, k): \mathbb{R}^n \times \mathbb{R}_+^p \times \mathbb{R}^q \times \mathbb{R}_{++} \rightarrow \mathbb{R}$ aşağıdaki gibi tanımlanır;

$$F(x, u, v, k) = f_0(x) - k^{-1} \sum_{i=1}^p u_i \ln(k f_i(x) + 1) - \sum_{j=1}^q v_j g_j(x) + k/2 \sum_{j=1}^q g_j^2(x) \quad (5.6)$$

$x \in \text{int } \Omega_k$ ve $\Omega_k = \{x: f_i(x) \geq -k^{-1}, i = 1, \dots, p\}$. Eğer (5.4)'deki tamamlayıcı yapay değişken_koşulu (x^*, u^*, v^*) noktasında sağlanıyorsa böylece herhangi bir $k > 0$ için:

$$(P1) \quad F(x^*, u^*, v^*, k) = f_0(x^*),$$

$$(P2) \quad \nabla_x F(x^*, u^*, v^*, k)$$

$$\begin{aligned}
&= f_0(x^*) - \sum_{i=1}^p \frac{u_i^*}{k f_i(x^*) + 1} \nabla f_i(x^*) - \sum_{j=1}^q (v_j^* - k g_j(x^*)) \nabla g(x^*) \\
&= f_0(x^*) - \sum_{i=1}^p u_i^* \nabla f_i(x^*) - \sum_{j=1}^q v_j^* \nabla g(x^*) = 0
\end{aligned}$$

$$(P3) \nabla_{xx}^2 F(x^*, u^*, v^*, k)$$

$$= \nabla_{xx}^2 L(x^*, u^*, v^*) + k \nabla f^T(x^*) U^* \nabla f(x^*) + k \nabla g^T(x^*) \nabla g(x^*).$$

Burada $\nabla f(x)$ ve $\nabla g(x)$ sırasıyla f ve g vektör fonksiyonlarının Jakobiyen matrisleridir. U^* bir diyagonal matristir.

5.3 Gradyant Tabanlı Yaklaşım

Biz burada yukarıda (5.6) olarak verilen fonksiyonu yalnızca eşitlik kısıtları için kullandık;

$$F(x, v, k) = f_0(x) - \sum_{j=1}^q v_j g_j(x) + k/2 \sum_{j=1}^q g_j^2(x) \quad (5.7)$$

Dinamik sistemimiz;

$$\frac{dx}{dt} = -\nabla F_x = -\nabla f_0(x) + \sum_{j=1}^q v_j \nabla g_j(x) - k \sum_{j=1}^q g_j(x) \nabla g_j(x) \quad (5.8)$$

v Lagrange çarpanı, $k > 0$ bariyer parametresidir. Denklem (5.8) optimizasyon problem için sınır dinamik modelini tanımlayan diferansiyel denklem sistemidir. Bu ayrıca yapay sinir ağı modelinin öğrenme kuralıdır.

Problem 5.3.1 (Problem No: 216, [49]):

$$\begin{aligned} \min \quad & f(x) = 100(x^2 - y)^2 + (x-1)^2 \\ \text{s.t.} \quad & h(x) = x(x-4) - 2y + 12 = 0 \end{aligned}$$

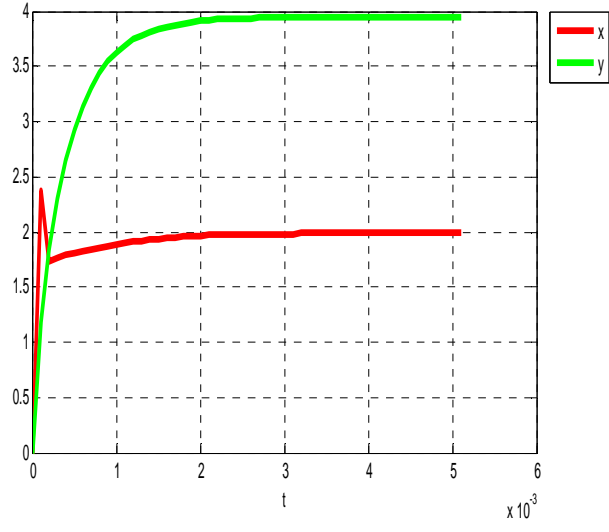
Optimizasyon problemi ele alınsın. Verilen başlangıç değerleri $(x_0, y_0) = (0, 0)$ şeklinde olup ulaşılabilecek optimum değer $(x^*, y^*) = (2, 4)$ şeklindedir. Amaç fonksiyonunu ve eşitlik kısıtını (5.7) denkleminde yazdığımızda problemimiz;

$$\begin{aligned} F(x, y, v, k) = & 100(x^2 - y)^2 + (x-1)^2 - v[x(x-4) - 2y + 12] \\ & + k/2[x(x-4) - 2y + 12]^2 \end{aligned} \quad (5.9)$$

kısıtlamasız bir optimizasyon problemine dönüşür. Buradan da bu (5.9) fonksiyonun gradyanının tersi alınarak dinamik sistem aşağıdaki gibi oluşturulur,

$$\begin{aligned} \frac{dx}{dt} = & -400(x^2 - y) - 2(x-1) + v(2x-4) - k[x(x-4) - 2y + 12](2x-4) \\ \frac{dy}{dt} = & 200(x^2 - y) - 2vy + 2k[x(x-4) - 2y + 12]. \end{aligned} \quad (5.10)$$

Verilen başlangıç koşulları ile birlikte Euler yöntemini kullanılarak (5.10) dinamik sistemi çözülür. Şekil 5.1 de ele alınan problemin karar değişkenlerinin çözüm yörüngesi Matlab programı yardımıyla elde edilmiştir. Burada adım sayısı $h = 0.0001$, Lagrange çarpanı $v = 50$ ve bariyer parametresi $k = 500$ olarak alınmıştır.



Şekil 5.1: Örnek 5.3.1 de $x(t)$ 'nin çözüm yörüngesi

6. HİPERBOLİK CEZA METODU İLE GRADYANT TABANLI YAKLAŞIM

Eşitsizlik kısıtlı genel doğrusal olmayan programlama problemlerini aşağıdaki gibi tanımlaya biliriz [50]:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geq 0, \quad i = 1, \dots, m \end{aligned} \quad (6.1)$$

$f: \mathbb{R}^n \rightarrow \mathbb{R}$ ve $g_i(x): \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$.

Yukarıdaki (6.1) problemini çözmek amacıyla, hiperbolik ceza metodu ilk olarak Xavier [51] tarafından sunulmuştur.

$$\min \quad f(x) + \sum_{i=1}^m \bar{P}(g_i(x)) \quad (6.2)$$

(6.2) kısıtlamasız problemde ikinci terim ceza terimi olup (6.1) probleminin kısıtlarını içermektedir.

Bir çok araştırmacı ceza metodları üzerinde teorik ve hesaba dayalı çalışmalar geliştirmişlerdir, bunlar için de Fiacco ve McCormick [52], Lootsma [53], Ryan [54], Fletcher [55] ve Wright [56] gösterilebilir. Ceza metodlarının genel özellikleri ve karakteristiği ders kitaplarında da bulunabilir, örneğin Minoux [57], Fletcher [58] ve Luenberger [59] gibi.

6.1 Hiperbolik Ceza

Hiperbolik ceza metodu problem (6.1)'deki eşitsizlik kısıtlarına sahip genel doğrusal olmayan programlama problemlerinin çözümünde kullanılır [50].

Hiperbolik ceza metodundaki ceza fonksiyonu aşağıdaki gibi kabul edilir;

$$P(y, \alpha, \tau) = -\left(\frac{1}{2} \tan \alpha\right) y + \sqrt{\left(\frac{1}{2} \tan \alpha\right)^2 y^2 + \tau^2} \quad (6.3)$$

$\alpha \in [0, \pi/2]$ ve $\tau \geq 0$.

Alternatif olarak ceza fonksiyonu daha uygun bir forma aşağıdaki gibi dönüştürülebilir;

$$P(y, \lambda, \tau) = -\lambda y + \sqrt{\lambda^2 y^2 + \tau^2} \quad (6.4)$$

$\lambda \geq 0$ ve $\tau \geq 0$.

6.2 Hiperbolik Ceza Algoritması

Problem (6.1)'in çözümü değiştirilmiş amaç fonksiyonunun minimizasyonu ile tanımlı alt problemlerin bir dizisinin çözümü aracılığıyla elde edilmiştir;

$$F(x, \lambda^k, \tau^k) = f(x) + \sum_{i=1}^m P(g_i(x), \lambda^k, \tau^k) \quad (6.5)$$

Hiperbolik ceza algoritması

1. Adım $k=0$ olsun. x^0 , $\lambda^1 > 0$ ve $\tau^1 > 0$ başlangıç değerleri alınır.
2. Adım $k : k+1$ olsun. Kısıtsız minimizasyon problemin çözümü:

$$\min_x F(x, \lambda^k, \tau^k)$$

x^{k-1} başlangıç noktasından bir ara optimal x^k noktası elde ediliyor.

3. Adım Uygunluk testi:

Eğer x^k uygun nokta değilse 4. Adımı uygula

diğer durumda 5. Adımı uygula

4. Adım λ parametresi üzerinde artış:

$$\lambda_i^{k+1} = r\lambda_i^k, \quad r > 1$$

2. Adıma git

5. Adım τ parametresi üzerinde azalma:

$$\tau_i^{k+1} = q\tau_i^k, \quad 0 < q < 1$$

2. Adıma git

Alt problemlerin dizisi algoritmanın iki farklı evresi için de “ λ ” ve “ τ ” parametrelerinin kontrol edilen değişkenleri ile elde edilir. İlk olarak λ parametresi artarken, uygun bölge dışındaki ceza içinde kayda değer bir artışa sebep olur. Bu süreç uygun bir nokta elde edene kadar devam eder. Bu noktadan sonra, λ sabit kalır ve τ değeri gittikçe azalır. İlk aşamada (λ parametresinin çalışması ile karakterize edilmiştir) hiperbolik ceza dış ceza metoduna benzer biçimde davranır. İkinci aşamada (τ parametresinin çalışması ile karakterize edilir) buradaki davranış ise iç ceza metoduna daha yakındır [50].

6.3 Gradyant Tabanlı Yaklaşım

Burada (6.5) hiperbolik ceza fonksiyonun ele alarak optimizasyon problemlerinin çözümü için kullanacağımız dinamik sistem aşağıdaki gibi tanımlanır,

$$\begin{aligned} \frac{dx}{dt} &= -\nabla f(x) - \sum_{i=1}^m \nabla P(g_i(x), \lambda^k, \tau^k) \\ \frac{d\lambda}{dt} &= \lambda(r-1), \quad r > 0 \\ \frac{d\tau}{dt} &= \tau(q-1), \quad 0 < q < 1 \end{aligned} \tag{6.6}$$

Denklem (6.6) optimizasyon problem için sinir dinamik modelini tanımlayan diferansiyel denklem sistemidir. Bu ayrıca yapay sinir ağı modelinin öğrenme kuralıdır.

Problem 6.3.1 (Problem No: 12, [60]) :

$$\begin{aligned} \min \quad & f(x) = 0.5x_1^2 + x_2^2 - x_1x_2 - 7x_1 - 7x_2 \\ \text{s.t.} \quad & \\ & g(x) = 25 - 4x_1^2 - x_2^2 \geq 0 \end{aligned} \tag{6.7}$$

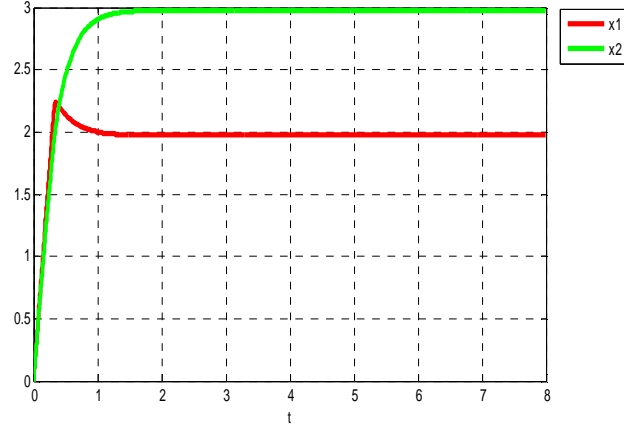
optimizasyon problemi ele alalım. Verilen başlangıç değerleri $x_0 = (0,0)$ şeklinde olup ulaşılabilecek optimum değer $x^* = (2,3)$ şeklindedir. Amaç fonksiyonunu ve eşitsizlik kısıtını (6.5) hiperbolik ceza fonksiyonunda yerine yazdığımızda;

$$\begin{aligned} F(x, \lambda, \tau) = & 0.5x_1^2 + x_2^2 - x_1x_2 - 7x_1 - 7x_2 - \lambda(25 - 4x_1^2 - x_2^2) \\ & + \sqrt{\lambda^2(25 - 4x_1^2 - x_2^2)^2 + \tau^2} \end{aligned}$$

kısıtlamasız optimizasyon problem elde edilir. (6.6) gradyant tabanlı dinamik sistemi yardımı ile çözüm sistemi aşağıdaki gibi oluşturulur;

$$\begin{aligned} \frac{dx_1}{dt} = & -x_1 + x_2 + 7 - 8\lambda x_1 + 8\lambda^2 x_1(25 - 4x_1^2 - x_2^2) / \sqrt{\lambda^2(25 - 4x_1^2 - x_2^2)^2 + \tau^2} \\ \frac{dx_2}{dt} = & -2x_2 + x_1 + 7 - 2\lambda x_2 + 2\lambda^2 x_2(25 - 4x_1^2 - x_2^2) / \sqrt{\lambda^2(25 - 4x_1^2 - x_2^2)^2 + \tau^2}. \end{aligned}$$

Verilen başlangıç koşulları ve Euler yöntemi kullanılarak ele alınan optimizasyon problemi (6.7) çözülür. Şekil 6.1 de problemim çözüm yörüngesi Matlab programı yardımıyla elde edilmiştir. Burada $h = 0.00001$, $q = 0.01$, $\lambda = 1$ ve $\tau = 1$ olarak alınmıştır.



Şekil 6.1: Örnek 6.3.1 de $x(t)$ 'nin çözüm yörüngesi

Problem 6.3.2 (Problem No:337, [49]):

$$\begin{aligned}
 \min \quad & f(x) = 9x_1^2 + x_2^2 + 9x_3^2 \\
 \text{s.t.} \quad & g_1(x) = x_1x_2 - 1 \geq 0 \\
 & g_2(x) = x_2 - 1 \geq 0 \\
 & g_3(x) = 1 - x_3 \geq 0
 \end{aligned} \tag{6.8}$$

optimizasyon problemini ele alalım. Verilen başlangıç değeri $x_0 = (1,1,1)$ şeklinde olup ulaşılabilecek optimum değer $x^* = (0.5774, 1.732, -0.2026E-05)$ şeklindedir. Amaç fonksiyonunu ve eşitsizlik kısıtını (6.5) hiperbolik ceza fonksiyonunda yerine yazdığımızda;

$$\begin{aligned}
 F(x, \lambda, \tau) = & 9x_1^2 + x_2^2 + 9x_3^2 - \lambda(x_1x_2 - 1) - \lambda(x_2 - 1) - \lambda(1 - x_3) \\
 & + \sqrt{\lambda^2(x_1x_2 - 1)^2 + \tau^2} + \sqrt{\lambda^2(x_2 - 1)^2 + \tau^2} + \sqrt{\lambda^2(1 - x_3)^2 + \tau^2}
 \end{aligned}$$

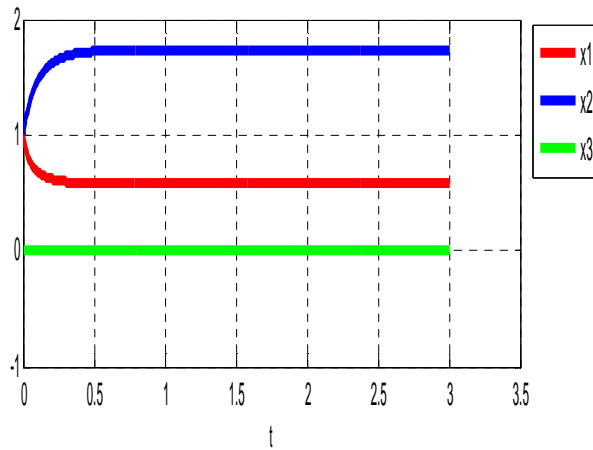
kısıtlamasız optimizasyon problemi elde edilir ve buradan da bu fonksiyonun gradyanının tersi alınarak (6.6) denklemi yardımı ile dinamik sistem aşağıdaki gibi oluşturulur;

$$\frac{dx_1}{dt} = -18x_1 + \lambda x_2 - \lambda^2 x_2 (x_1 x_2 - 1) / \sqrt{\lambda^2 (x_1 x_2 - 1)^2 + \tau^2}$$

$$\frac{dx_2}{dt} = -2x_2 + \lambda x_1 - \lambda^2 x_1 (x_1 x_2 - 1) / \sqrt{\lambda^2 (x_1 x_2 - 1)^2 + \tau^2} + \lambda - \lambda^2 (x_2 - 1) / \sqrt{\lambda^2 (x_2 - 1)^2 + \tau^2}$$

$$\frac{dx_3}{dt} = -18x_3 - \lambda + \lambda^2 x_1 (1 - x_3) / \sqrt{\lambda^2 (1 - x_3)^2 + \tau^2}$$

Verilen başlangıç koşulları Euler yöntemi kullanılarak(6.8) optimizasyon problem çözülür. Şekil 6.2 de ele alınan problemin çözüm yörüngesi Matlab programı yardımıyla elde edilmiştir. Burada $h = 0.0001$, $q = 0.01$, $r = 1000$, $\lambda = 100$ ve $\tau = 0.1$ olarak alınmıştır.



Şekil 6.2: Örnek 6.3.2 de $x(t)$ 'nin çözüm yörüngesi

7. SONUÇLAR

Optimizasyon genel anlamda modelleme ve çözümlene gibi iki bileşen olarak nitelendirilebilir. Modelleme problemin matematiksel olarak ifade edilmesi, çözümlene ise bu matematiksel modelin en iyi çözümlünün elde edilmesidir. Bu tezde öncelikli olarak optimizasyon problemlerinin optimal sonucu elde edebilmek için gerekli koşullar ve optimizasyon problemleri tanıtılmıştır. İkinci bölümde YSA'lar hakkında literatürde ki bilgilere yer verilmiştir. Daha sonra optimizasyon ve YSA'ların ortak kullanım alanlarına değinilmiştir.

Son iki bölümde değıştırilmiş bariyer ve genişletilmiş Lagrangian metodu yardımıyla eşitlik kısıtlarına sahip optimizasyon problemlerinin bazı sınıfları için bir dinamik sistem yaklaşımı uygulanmıştır. Bununla beraber hiperbolik ceza metodu yardımı ile eşitsizlik kısıtlarına sahip NLP problemleri için bir dinamik sistem yaklaşımı oluşturulmuştur. Ele alınan optimizasyon problemleri için modellenen dinamik sistemler Euler yöntemi kullanılarak optimal çözümleri araştırılmıştır. Yapılan uygulamalarda dinamik sistemin denge noktasının optimizasyon probleminin optimal çözümlüne karşılık geldiği görülmüştür.

Elde edilen dinamik sistemlerin kararlılığına bakılarak bu dinamik sistemler sinir dinamiği olarak gösterilip, optimal çözümlene ulaşılabilir.

8. KAYNAKLAR

- [1].Gottfried, B.S., Weisman, J., *Introduction to Optimization Theory*, New Jersey, Prentice Hall Inc.,(1973).
- [2].Clapham, C., *Coincise Dictionary of Mathematics*, 2 nd Edition, New York, Oxford University Press, (1996).
- [3].Cooke, R., *The History of Mathematics: A Brief Course*, Canada, John Wiley & Sons Inc., (1997).
- [4].Gass, S. I., "Great moments in history", *OR/MS Today*, 29 (5), (2002).
- [5].Çetin, E., Stokastik Programlama Yöntemiyle Bir Portföy Optimizasyonu Modelinin Geliştirilmesi, Yayımlanmamış Doktora Tezi, İstanbul, *İ.Ü. Sosyal Bilimler Enstitüsü*, Sayısal Yöntemler Bilim Dalı, (2004).
- [6].Harvey, R. L., *Neural Network Principles*, Prentice-Hall, Inc., New Jersey, (1994).
- [7].Karna, K.N., David, M.B., "An artificial neural networks tutorial part:1-basics", *Neural Networks*, 1(1), 5-23, (1989).
- [8].Öztemel, E., *Yapay Sinir Ağları*, Papatya Yayıncılık,İstanbul, (2006).
- [9].Bodily, S. E. etc, *Quantitative Business Analysis: Text and Cases*, McGraw-Hill, (1998).
- [10].Chong, K. P. E. and Zak, H. S., *An Introduction to Optimization*, third edition, New Jersey: John Wiley and Sons, (2008).
- [11].Luenberger D. G. and Ye Y, *Linear and Nonlinear Programming*, third edition. New York, Springer, (2008).
- [12].McCulloch, W. S. and Pitts, W. A., "A logical calculus of the ideas immanent in nervous activity", *Bulletin of Mathematics and Biophysics*, 5, 115-133, (1943),
- [13].Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychoanalytic Review*, 65, 386-408, (1958).
- [14].Widrow, B. and Hoff, M. E., "Adaptive switching circuits", *WEST-CON Convention*, Record Part IV, 1960, 96-104, (1960).

- [15]. Grossberg, S.A., "Contour enhancement, short term memory, and consultancies in reverberating neural networks", *Studies in Applied Mathematics*, 52(3), 213-257, (1973).
- [16]. Carpenter, G. A. and Grossberg, S. A., "ART2: self-organization of stable category recognition codes for analog input patterns", *Applied Optics*, 26(3), 4919-4930, (1987).
- [17]. Fukushima, K., "Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological Cybernetics*, 36, 193-202, (1980).
- [18]. Fukushima, K., "Neocognitron: a new algorithm for pattern recognition of deformations and shift in position", *Pattern Recognition*, 15, 455-469, (1982).
- [19]. Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities", *Proceedings of the National, Academy of Sciences*, 79, 2554-2558, (1982).
- [20]. Hopfield, J.J., "Neurons with graded response have collective computational properties like those of two state neurons", *Proc. Natl. Acad. Sci.*, 81, 3088-3092, (1982).
- [21]. Broomhead, D.S., Lowe, D., *Radial basis-functions, multi-variable functional interpolation and adaptive networks*, Royal signals and radar establishment memorandum , (1988).
- [22]. Arslan, O., Kurt, O. ve Konak, H., *Yapay Sinir Ağlarının Jeodezide Uygulamaları Üzerine Öneriler*, (2007).
- [23]. Miyamoto, H., Kawato, M., Setoyama, T. and Suzuki, R., "Feedback-error-learning neural network for trajectory control of a robotic manipulator", *Neural Networks*, 1, 251-265, (1988).
- [24]. Adeli, H., and Park, Hyo Seon, "A neural dynamics model for structural optimization-theory", *Computer and Structures*, 57(3), 383-390, (1995).
- [25]. Hertz J. , Krogh A. and Palmer, R., *Introduction to the Theory of Neural Networks*, Addison Wesley, (1991).
- [26]. Wen, U. P., Lan, K. M., Shih, H. S., A review of Hopfield neural networks for solving mathematical programming problems, 198 No:3, 675-687, (2009).
- [27]. Hopfield, J. J. and Tank, D. W., "Simple neural optimization networks: an A/D converter, signal decision circuit, and a linear programming circuit". *IEEE Trans. Circuits Systems*, 33, 533-541, (1986).

- [28]. Kennedy, M. P., Chua, L. O., “Neural networks for nonlinear programming”, *IEEE Transportation Circuits System* 35, 554-562, (1988).
- [29]. Rodríguez-Vázquez, A., Domínguez-Castro, R., Rueda, A., Huertas, J. L., Sánchez-Sinencio, E., “Switched-capacitor neural networks for linear programming”, *Electronics Letters* 24, 496-498, (1988).
- [30]. Rodríguez-Vázquez, A., Domínguez-Castro, R., Rueda, A., Huertas, J. L., Sánchez-Sinencio, E., “Nonlinear switched-capacitor ‘neural networks’ for optimization problems”, *IEEE Transactions on Circuits systems* 37, 384-397, (1990).
- [31]. Zhang, S., Constantinides, A. G., “Lagrange programming neural networks”, *IEEE Transactions on Circuits Systems* 39, 441-452, (1992).
- [32]. Cichocki, A., Unbehauen, R., *Neural Networks for optimization and Signal Processing*, John Wiley, New York, (1993).
- [33]. Xia, Y., Feng, G., Wang, J., “A primal-dual neural network for online resolving constrained kinematic redundancy in robot motion control”, *IEEE Transactions on Systems Man and Cybernetics Part B–Cybernetics* 35, 54-64, (2005).
- [34]. Tank, D. W., Hopfield, J. J., “Simple ‘neural’ optimization networks: An A/D converter, signal decision network and a linear programming circuit”, *IEEE Transactions on Circuits Systems* 33, 533-541, (1986).
- [35]. Kennedy, M. P., Chua, L. O., “Neural networks for nonlinear programming”, *IEEE Transportation Circuits System* 35, 554-562, (1988).
- [36]. Maa, C. Y., Shanblatt, M. A., “Linear and quadratic programming neural network analysis”, *IEEE Transaction on Neural Networks*, 3, 380-394, (1992).
- [37]. Bazaraa, M.S., Sherali, H. D., Shetty, C. M., *Nonlinear Programming Theory and Algorithms*, third ed. John-Wiley, New York, (2006).
- [38]. Effati, S., Baymain, M., “A new nonlinear neural network for solving convex nonlinear programming problems”, *Applied Mathematics and Computation* 168, 1370-1379, (2005).
- [39]. Gill, E. E., Murray, W., Wright, M. H., *Practical Optimization*, Academic, London, (1981).
- [40]. Shih, H. S., Wen, U. P., Lee, E. S., Lan, K. M., Hsiao, H. C., “A neural network approach to multiobjective and multilevel programming problems”, *Computers and Mathematics with Applications*, 48, 95-108, (2004).

- [41]. Ham, F. M., Kostanic, I., *Principles of Neurocomputing for Science and Engineering*, McGraw-Hill, New York, (2001).
- [42]. Gong, D., Gen, M., Yamazaki, G., Xu, W., “Lagrangian ANN for convex programming with linear constraints”, *Computers and Industrial Engineering*, 32, 429-443, (1997).
- [43]. Wu, A.I., Tam, P.K.S., “A neural network methodology and strategy of quadratic optimization”, *Neural Computing and Applications*, 8, 283-289, (1999).
- [44]. Polyak, R., “Modified barrier functions (theory and methods)”, *Mathematical Programming*, 54, 177-222, (1992).
- [45]. Hestenes, M.R., “Multiplier and gradient methods”, *Journal of Optimization Theory and Applications*, 4, 303-320, (1969).
- [46]. Powell, M.J.D., “A method for nonlinear constraints in minimization problems”, in *Optimization*, R. Fletcher (Ed.), *Academic Press: London, New York*, 283-298, (1969).
- [47]. Rockafellar, R. T., “The multiplier method of Hestenes and Powell applied to convex programming”, *JOTA*, 12, 555-562, (1973).
- [48]. Goldfarb, D., Polyak, R., Scheinberg, K., Yuzefovich, I., “A modified barrier-augmented Lagrange method for constrained minimization”, *Computational Optimization and Applications*, 14, 55-74, (1999).
- [49]. Schittkowski, K., *More Test Examples for Nonlinear Programming Codes*, Springer Verlag, (1987).
- [50]. Xavier, A. E., “Hyperbolic penalty: a new method for nonlinear programming with inequalities”, *International Transactions in Operational Research*, 8, 659-671, (2001).
- [51]. Xavier, A. E., Penalização Hiperbolica-Um Novo Metodo para Resolucao de Problemas de Otimização. *MSc Thesis*, Federal University of Rio de Janeiro/COPPE, Rio de Janeiro, (1982).
- [52]. Fiacco, A.V., McCormick, G. P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley&Sons, New York, (1968).

- [53]. Lootsma, F. A., "A Survey of Methods for Solving Constrained Minimization Problems Via Unconstrained Minimization", In Lootsma, F. A. (Ed.) Numerical Methods for Non-Linear Optimization, *Academic Press*, London, 313-395, (1972).
- [54]. Ryan, D. M., "Penalty and Barrier Functions", In Gill, P. E., Murray, W., Eds., Numerical Methods for Constrained Optimization, *Academic Press*, London and New York, 175-190, (1974).
- [55]. Fletcher, R., "Penalty Functions", in Bachen, A., Grotchel, M., Komte, B. (Eds.), *Mathematical, Programing-The State of Art*, Springer Verlag, Berlin, 87-144, (1983).
- [56]. Wright, M. H., "Interior Methods for Constrained Optimization", *Acta Numerica*, 341-407, (1991).
- [57]. Minoux, M., *Mathematical Programming Theory and Algorithms*, John Wiley&Sons, Chichester, (1986).
- [58]. Fletcher, R., *Practical Methods of Optimization*, 2nd edn. John Wiley&Sons, Chichester, (1987).
- [59]. Luenberger, D., *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Menlo Park, (1984).
- [60]. Hock, W., Schittkowski, K., *Test Examples for Nonlinear Programming Codes*, Berlin, Springer Verlag, (1981).