

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY SİNİR AĞLARININ
GEMİ YAPI PROBLEMLERİNE UYGULANMASI**

**YÜKSEK LİSANS TEZİ
Müh. Serdar Aytekin KÖROĞLU
508051014**

**Tezin Enstitüye Verildiği Tarih : 7 Mayıs 2007
Tezin Savunulduğu Tarih : 12 Haziran 2007**

Tez Danışmanı : Prof.Dr. Ahmet ERGİN

**Diğer Jüri Üyeleri : Yrd.Doç.Dr. Ertekin BAYRAKTARKATAL
Dr. Meriç TEMİZKAYA**

HAZİRAN 2007

ÖNSÖZ

Bu tezin hazırlanmasında pek çok kişinin emeği geçmiştir. Öncelikle, değerli öngörüler, cesaret ve güven veren desteğiyle bu çalışmaya en önemli katkıyı yapan hocam Prof. Dr. Ahmet Ergin teşekkürlerimi sunarım. Doğadan ilham alan hesaplama konularıyla ilgili altyapımın oluşmasını sağlayan Yrd. Doç. Dr. Şima Etaner Uyar'a ve kısa süre içerisinde yapay sinir ağları konusunda çok kritik yardımlarda bulunan Yrd. Doç. Dr. Neslihan Serap Şengör'e de teşekkürü bir borç bilirim. Ayrıca çok değerli dostlarıma ve bugüne kadar attığım her adımda arkamda olan aileme de şükranlarımı sunarım.

Mayıs 2007

Serdar Aytekin KÖROĞLU

İÇİNDEKİLER

ÖNSÖZ	ii
TABLO LİSTESİ	iv
ŞEKİL LİSTESİ	v
ÖZET	vi
ABSTRACT	vii
1. GİRİŞ	1
1.1 Tasarım ve Optimizasyon	2
1.2 Büyük Ölçekli Optimizasyon ve Genetik Algoritmalar	2
1.3 Yapay Sinir Ağlarının Rolü	3
1.4 Diğer Çalışmalar	5
2. DENEY TASARIMI VE CEVAP YÜZEYİ YÖNTEMİ	6
2.1 Giriş	6
2.2 En Küçük Kareler Yöntemiyle Çok Değişkenli Regresyon	6
2.3 İki Seviyeli Faktöriyel Tasarımı	7
2.3.1 Modeli Oluşturacak Ana Etkilerin Belirlenmesi	9
2.3.2 Model ve Model Terimleri İçin Hipotez Testi	10
2.3.3 Artık Analizi	11
2.3.4 Cevap Değişken Dönüşümü	13
2.4 Merkezi Kompozit Tasarımı	13
2.5 Kesirsel Faktöriyel Tasarımı	14
2.5.1 Yarım 2^k Tasarımı	14
2.6 Ek Bilgi	15
3. YAPAY SİNİR AĞLARI	17
3.1 Öğrenme	19
3.2 Evrensel Yaklaşım ve Kolmogorov Teoremi	19
3.3 Geri Yayınım Algoritması	20
3.3.1 En-Dik İniş Algoritması	21
3.3.2 Levenberg-Marquardt Algoritması	24
3.4 Giriş Çıkış Değerleri Normalizasyonu	25
3.5 Genelleştirme	25
3.5.1 Erken Durdurma	26
3.5.2 Ağ Budama	27
4. STİFNERLİ BİR PANELİN LİNEER BURKULMA MODELLEMESİ	28
4.1 Problemin Tanımı	29
4.2 Deney Tasarımı ile Modelleme	31
4.3 Yapay Sinir Ağlarıyla Modelleme	33
4.4 Karşılaştırmalı Performans	36
5. GELECEK ÇALIŞMALAR	38
KAYNAKLAR	39
ÖZGEÇMİŞ	41

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1: Etki Hesabı İçin İşaret Tablosu.....	8
Tablo 2.2: Yarım 2k Tasarım İşaret Tablosu	15
Tablo 4.1: Dizayn Parametreleri ve Tanım Aralıkları.....	30
Tablo 4.2: ANOVA Tablosu	32

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1: 2^2 Faktöriyel Tasarımı	8
Şekil 2.2: Etki Normal Olasılık Grafiği	9
Şekil 2.3: Artık Normal Olasılık Grafiği	12
Şekil 2.4: Cevap - Artık Grafiği	12
Şekil 2.5: İki Parametrelili Merkezi Kompozit Tasarımı.....	13
Şekil 3.1: Nöron Modeli	17
Şekil 3.2: İleri Beslemeli Ağ	18
Şekil 3.3: Hata geri besleme	22
Şekil 3.4: Aşırı-Uyum Problemi	26
Şekil 4.1: Stifnerli Panelin Geometrisi	29
Şekil 4.2: Örnek bir Burkulma Mod Şekli.....	30
Şekil 4.3: Yarı-Normal Grafik.....	31
Şekil 4.4: Artık Normal Olasılık Grafiği	33
Şekil 4.5: Veri Sayısının Performansa Etkisi	35
Şekil 4.6: Nöron Sayısının Performansa Etkisi	35
Şekil 4.7: Hata Performans Değerleri	36
Şekil 4.8: Örnek bir alt küme için gerçek ve tahmini değerler	37

ÖZET

Günümüzde, hızla gelişen teknolojik imkânların da etkisiyle, eskiye nazaran daha karmaşık ve daha büyük ölçekli problemler çözülebilir hale gelmiştir. Bu duruma iyi bir örnek olarak *genetik algoritmalar* yaklaşımının gemi yapılarının optimum tasarımı gibi çözümü zor problemlere uygulanmasını gösterebiliriz.

Genetik algoritmalar optimizasyon açısından güçlü bir yöntem olmasına rağmen, çözüme ulaşmak için mevcut en hızlı bilgisayarlar bile kimi zaman yetersiz kalmaktadır.

Buradan hareketle işlem yükünü hafifletmek üzere genetik algoritmaların performansına doğrudan etki eden analiz safhası için alternatif yöntemler kullanılmaktadır. Bu yöntemlerden olan yapay sinir ağları ve deney tasarımı yöntemleri sonlu elemanlar yöntemine alternatif olarak kullanılmak üzere modelleme açısından ayrıntılı olarak irdelenmiştir. Ayrıca kullanımları, örnek bir yapı problemi (stifnerli bir panelin burkulma hadisesinin modellenmesi) üzerinden uygulamalı olarak gösterilmiş ve sonuçları karşılaştırmalı olarak incelenmiştir.

APPLICATION OF ARTIFICIAL NEURAL NETWORKS TO SHIP STRUCTURAL DESIGN

SUMMARY

Nowadays, with the help of rapid improvements in technology, more complex and much bigger scale problems are become solvable with respect to past. As a good example, optimum design of ship structures, which is a very tough problem, can be solved with applying *genetic algorithms*.

Despite the fact that genetic algorithms are very powerful for optimization, even the fastest computers available remain inadequate for reaching the solution.

With this point in mind, to lessen the burden on genetic algorithms, some alternative methods are utilized for the analysis stage, which directly affect the performance. Two of these methods, namely artificial neural networks and design of experiments have been studied from the modeling point of view as alternatives to finite element method. In addition, the usage of the methods has been shown by applying them for a structural problem (modeling the buckling behavior of a stiffened panel) and finally results are comparatively analyzed.

1. GİRİŞ

Mühendislikte tasarım, bir ihtiyacı karşılamak veya mevcut bir çözümden daha iyisini yapmak adına yeni bir sistemi ortaya koymak olarak tanımlanabilir. Diğer yandan tasarım, çözüm alternatiflerini sınırlayan kısıtlar ve kalan alternatiflerden en iyi çözümün bulunması (optimizasyon) gibi unsurlar tarafından yönlendirilir.

Gemi tasarımında ise yukarıda bahsedilen genel prensipler aynen geçerli olup, ayrıca çok amaçlılık, büyük ölçekte ve yüksek karmaşıklıkta olması tasarım probleminin konvansiyonel yöntemlerle¹ çözümünü imkânsızlaştırmaktadır.

Son dönemde bilgisayar teknolojisinin hızlı ilerleyişiyle artık böyle büyük problemlerin çözümü için alternatif yöntemler kullanılabilmektedir. Bu yöntemlerden biri olan “Genetik Algoritmalar”, Darwin’in “Evrin Teorisi”nden hareketle bulunan ve büyük ölçekli, lineer olmayan hatta süresiz bir fonksiyonla tanımlı bir problemin optimizasyonunu (hem de konvansiyonel yöntemlerin aksine global olarak) gerçekleştirebilen bir yöntemdir.

Genetik algoritmaların böyle güçlü özelliklerinin yanı sıra bazı dezavantajlara da sahiptir. Bunlardan birisi, problem çözümünde hesapların yüksek hızla yapılmasına ihtiyaç duymasıdır ve bu da problemin analizinde kullanılacak yöntemlere bazı kısıtlamalar getirir. Örneğin “sonlu elemanlar yöntemi” büyük ölçekli ve karmaşık modellerin analizinde oldukça fazla zamana ihtiyaç duyar ve bu da optimizasyon çalışmasının makul sürelerde tamamlanmasını zorlaştırır.

Bu tez çalışmasında bu noktada bir çözüm olarak kullanılabilecek bir yaklaşım olan *yapay sinir ağlarıyla modelleme* uyarlanacak ve alternatif olarak kullanılabilecek olan *deney tasarımı ve cevap yüzey* yöntemine olan üstünlükleri uygulamalı olarak gösterilecektir.

¹ Optimizasyon ve optimizasyonda kullanılan konvansiyonel metotlarla ilgili detaylı bilgi için Arora(1989)’a başvurulabilir.

Fakat önce bu bölümde, tasarım, optimizasyon ve yapay sinir ağlarının genetik algoritmalarla optimizasyondaki rolü hakkında bir ön bilgilendirme yapılacaktır. 2. ve 3. bölümlerde deney tasarımı ve cevap yüzey metodolojisi, yapay sinir ağları ve çok katmanlı perseptron ağları konuları irdelenecek ve son olarak 4. bölümde gemi yapılarında sıkça kullanılan stifnerli bir panelin burkulma hadisesinin modellenmesi her iki yöntemle uygulaması anlatılacaktır.

1.1 Tasarım ve Optimizasyon

Bir ürünü tasarlamak için, görev tanımı, teknolojik imkânlar, çevre şartları, maliyet, ergonomi, estetik kaygılar, güvenlik, güvenilirlik gibi pek çok faktörün bir arada düşünülmesi gerekir. Bu faktörler kimi zaman eklenti olarak dâhil edilebildiği gibi kimi zaman diğer faktörler açısından bir uzlaşımı zorunlu kılar.

Böyle karmaşık bir yapıya sahip bu süreci modüllere ayırarak ve tekrarlamalı (iteratif) yaklaşımlarla bir derece olsun basitleştirme imkânı yaratılabilir. Modüllere ayırma, bağımsız alt sistemlerin varlığı ile mümkün olabilir. Bu bile global tasarıma etkileri düşünülmediği sürece o bölgedeki bir iyileştirmeye sınırlı kalır. Tekrarlamalı bir yaklaşımda ise tasarımdaki parametrelerde sınırlı değişiklikler yapıp, bu değişikliklerin olumlu etki yarattığı yönde çalışmayı yönlendirmekle gerçekleştirilir. Fakat bu yaklaşım, problem uzayında bir lokal arama girişimidir ve global optimum'a ulaşmak neredeyse mümkün değildir (şans eseri global optimum civarında bir noktadan başlanılmamış ise).

Öte yandan problemi matematiksel bir modelin optimizasyonu olarak değerlendirdiğimizde, model parametreleri çok değişik formlarda karşımıza çıkabilir. Mesela parametreler sayısal, niteliksel (kategorik) veya bir kümeye ait sonlu sayıda elemanın dizilimi(kombinatorial) şeklinde olabilir. Diğer yandan sayısal parametreler de reel, tamsayı veya ayrık elemanlardan oluşan belli bir kümeye ait olması gibi, konvansiyonel yaklaşımda farklı farklı yöntemlerin uygulanmasını gerektiren tiplerde olabilirler.

1.2 Büyük Ölçekli Optimizasyon ve Genetik Algoritmalar

Gemi tasarımı, içinde çok sayıda parametre, kısıt ve birden fazla hedef tanımı içerir. Problemin böylesine büyük bir ölçekte olması ve önceki bölümde bahsedilen

zorluklar nedeniyle genetik algoritmalarla optimizasyon daha etkili bir çözüm yöntemi olarak karşımıza çıkar.

Genetik algoritmalar, Darwin'in *doğal seleksiyonunda* olduğu gibi, ortamdaki en iyilerin ayakta kalması prensibine dayanan, stokastik bir arama yöntemidir. Bu yöntemde tıpkı gerçek dünyada olduğu gibi bir toplum bulunmaktadır. Toplumdaki her birey farklı birer tasarım noktasını temsil eder. Bu bireylerden güçlü olanlar tespit edilir ve kendi aralarında birey tiplerine uygun operatörler yardımıyla yeni bireyler oluşturulur. Zayıf bireyler ise toplumdan silinir. Böylelikle her yeni nesille en iyi duruma daha da yaklaşılma imkânı yaratılmış olur.

Yöntemin avantajlarından biri, çözümün başlangıcında problem uzayında rasgele dağılan noktalarla (bireylerle) başlanıp, birey sayısı kadar farklı noktada da sürdürüldüğü için global optimuma ulaşım için uygun bir yöntemdir.

Bununla birlikte bireylerin kodlanması çok çeşitli parametre tipleri için uyarlanabilir. Bu da aynı yöntemle, parametre açısından heterojen yapıda problemlere çözüm için tek bir adımda arama yapılabilmesi demektir. Hatta kodlamada uygun bir strateji kullanılarak kısıtlı optimizasyon problemleri için ek bir uygulama gerektirmeden arama yapılabilir.

Daha birçok avantaja sahip olan² genetik algoritmaların bazı dezavantajları da vardır. Şimdi, bu tez çalışmasına konu olan ve bu dezavantajlardan birini gidermek için geliştirilen çözümden bahsedilecektir.

1.3 Yapay Sinir Ağlarının Rolü

Genetik algoritmalar, bireyleri birbirlerine kıyaslamak için, analiz neticesinde elde edilen değerlere belli bir performans ölçütü atar. Analizler gemi yapısı ile ilgili bir problemde sonlu elemanlar gibi bir yöntemle yapılabilir. Fakat her nesilde (toplumdaki bireylerin yenilenmesi) birey sayısı kadar analiz yapılması, çözüm boyunca on binlerce analizin yapılması anlamına gelir. Sonlu elemanlar yönteminden, günümüz teknolojisiyle, bu kadar çok analiz için kabul edilebilir süreler içinde sonuç vermesi beklenemez.

² Bu çalışmada genetik algoritmalarla daha detaylı olarak bahsedilmeyecek olup, okurun Eiben ve Smith (2003), Haupt ve Haupt (2004) ve Mitchell (1998) in yayınlarına başvurmaları tavsiye edilir.

Analiz, kıyaslamada kullanıldığı için %100 doğrulukta bir performans değeri (kaldı ki sonlu elemanlar yönteminde de bu elde edilemez) gerekli değildir. Yani performans ölçütü toplumun o anki bireylerine bağıl olarak kullanılır.

Dolayısıyla genetik algoritmaların işleyişinde önemli bir darboğaz olan performans fonksiyonu için sonlu elemanlar yöntemine alternatif, daha hızlı çalışan ve hata büyüklüğü optimizasyon çalışmasında önemli bir problem teşkil etmeyecek düzeyde olan yöntemlerin değerlendirilmesi gerekmektedir.

Deney tasarımı, istatistiksel metotların yardımıyla, deney sayısının mümkün olduğunca az tutularak bir model oluşturulmasını sağlar. Öncelikle tasarımın seçili parametrelerinin belirli limit değerlerinin kombinasyonlarıyla yapılan deney neticeleri kullanılarak bir varyans analizi (ANOVA) yapılır. Buradan yola çıkarak parametrelerin ve parametreler arası ilişkilerin modele ne kadar etki ettiği bilgisine ulaşılır. Daha sonra etkin parametreler ve onların etkileşimlerinin lineer bir denklemin terimleri olarak alındığı bir regresyon hesabı yapılır ve modeli temsil eden bir fonksiyona ulaşılmış olur.

Deney tasarımından farklı olarak “yapay sinir ağları”nda belirli değerler üzerinden değil, rasgele seçilen değerler (bu çalışmada kullanılan) veya geçmişte elde edilen mevcut değerler aracılığıyla model oluşturulur. Regresyon hesabına benzer fakat içinde bulundurduğu lineer olmayan terimler ve en uygun katsayıların aranmasında kullanılan farklı yaklaşımlar yöntemi daha güçlü kılar. Kısaca, giriş değerlerinin katsayılarla ağırlıklandırılıp, lineer olmayan bir transfer fonksiyonundan geçtiği elemanların oluşturduğu bir ağ yapısıdır. Burada en önemli iki parametre ağ yapısı (eleman sayısı, birbiri arasındaki bağlantı yapısı vs.) ve elemanlardaki katsayılardır.

Deney tasarımı lineer bir regresyon olduğundan ötürü, parametrelerin belirlenen limit değerleri arasındaki mesafenin, cevaba etkisinin lineer kalması için sınırlı tutulması gerekmektedir. Bu nedenle örneğin bir iyileştirme çalışmasında, tasarımda yapılacak değişikliğin sınırlı tutulması ve yöntemin adım adım uygulanması gerekmektedir. Yapay sinir ağlarında böyle bir uygulamaya gerek yoktur. Fakat modelin iyi oluşturulması için yeterli sayıda veri setinin mevcut olması gereklidir.

Bu çalışmada hem deney tasarımı hem de yapay sinir ağları örnek bir probleme uygulanmıştır. 2. ve 3. bölümlerde yöntemlerle ilgili tanıtıcı bilgiler verilecek olup,

4. bölümde örnek çalışmada kullanılan teknikler ve elde edilen sonuçlar gösterilecektir.

1.4 Diğer Çalışmalar

Literatüre baktığımızda, yapısal optimizasyon uygulamalarında bu tezde kullanılan her iki yöntemin de kullanıldığını görmekteyiz.

Kompozit stifnerli panellerin burkulma sonrası performansı açısından optimizasyonu probleminde, Lanzi ve Giavotto (2006) burkulma sonrası davranışın modellenmesi için geri-yayınım, radyal baz fonksiyonu ve Kriging yöntemlerini kullanmışlardır. Kullanılan yöntemlerin optimizasyon çalışmasına yakın katkılar sağladığı gözlenmiştir.

Todoroki ve Ishikawa (2003) lamine kompozit yapılarda dizilim sıralamasının burkulma açısından optimizasyonu için deney tasarımı yöntemini değerlendirmişlerdir.

Benzer bir çalışmada Abouhamze ve Shakeri (2006) lamine silindirik panellerin burkulma dayanıklılığı ve doğal titreşim açısından optimizasyonu için geri-yayınım ağını kullanmışlardır.

Waszczyszyn ve Bartczak (2002) da yine silindirik bir panelin optimizasyonu için geri-yayınım ağını değerlendirmiş olup bu çalışmada farklı olarak panel kompozit olmayıp, ayrıca geometrik açıdan kusurlu olarak hesaba dâhil edilmiştir.

Panellerin optimizasyonu dışında Ganguli (2002), bir helikopter rotorunun titreşim açısından optimizasyonunda titreşim davranışını, deney tasarımını temel alan merkezi kompozit tasarımı yöntemi ile modellemiştir.

2. DENEY TASARIMI VE CEVAP YÜZEYİ YÖNTEMİ

2.1 Giriş

Sanayide bir işlem veya bir ürünün çeşitli kalite ve performans özelliklerine birçok değişken etki eder. Etki eden bu değişkenlere “bağımsız değişken”, neticede elde edilen özelliğe de “cevap” adı verilir. Tasarımın iyileştirilmesi için bağımsız değişkenlerle cevap değişkenleri arasındaki ilişkinin modellenmesi şarttır. Deney tasarımı, minimum sayıda deney³ yapılarak bu ilişkilendirmenin gerçekleştirilmesini amaçlar. Bunun dışında seçilen bağımsız değişkenlerin modele ne kadar etki ettiğini istatistiksel metotlar aracılığıyla gerçekleştirip tasarımcıya, iyileştirmenin hangi değişkenlerde yapıldığında daha etkili olacağı hakkında fikir verir.

Deney tasarımı, temelde en küçük kareler yöntemiyle, modellenecek olayın seçili parametreleri için belirlenen minimum ve maksimum değerleri üzerinden çok değişkenli regresyon yapılmasına dayanır.

2.2 En Küçük Kareler Yöntemiyle Çok Değişkenli Regresyon

Birinci dereceden çok değişkenli bir cevap yüzeyini:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (2.1)$$

şeklinde gösterebiliriz. En küçük kareler yöntemi, bir olayla ilgili yapılan $n > k$ adet gözlemden elde edilen (x_{ij}, y_i) değerleriyle oluşturulan

$$y_i = \beta_0 + \sum_{j=1}^k \beta_j x_{ij} + \varepsilon_i \quad i = 1, 2, \dots, n \quad (2.2)$$

denklem sistemindeki ε_i hata değerlerinin karelerinin toplamını minimum yapacak şekilde β_j 'lerin tespit edilmesi ile gerçekleştirilir. Matris notasyonu ile

³ Deneyle, sayısal deneyler(simülasyon) kastedilmektedir.

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix} \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

gösterirsek $\boldsymbol{\beta}$,

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y} \quad (2.3)$$

den elde edilir (Myers and Montgomery, 2002). Eğer parametreler arası ilişkiler veya daha yüksek dereceden terimler de modelde kullanılmak isteniyorsa bu değerler elde edilen verilerden üretilir ve diğer parametreler gibi regresyon hesabına sokulur.

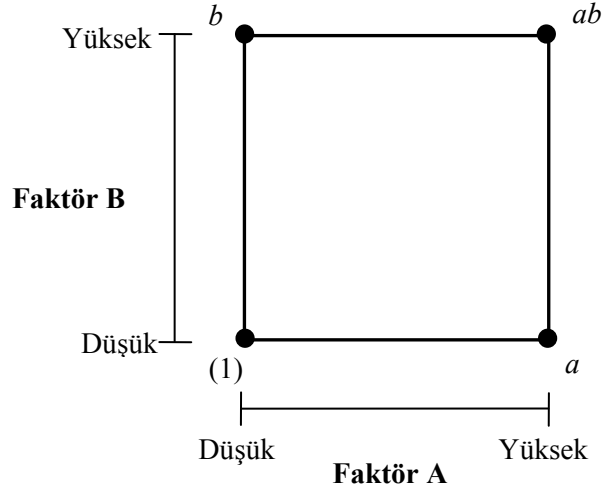
2.3 İki Seviyeli Faktöriyel Tasarımı

Parametreler arası ilişkilerin de kullanılabilmesi için *faktöriyel tasarım* adı verilen yöntem kullanılır. Sıklıkla parametrelerin ikişer seviyelerinin kombinasyonlarının kullanıldığı ve dolayısıyla 2^k adet deney yapılması gerektiğinden, 2^k faktöriyel tasarımı olarak da anılmaktadır.

2^k faktöriyel tasarımı, perdeleme adı verilen ve bir olaya etki eden faktörlerden ve onların etkileşimlerinden dominant olanların tespit edilmesi için kullanılmaktadır. Ayrıca bu çalışmada da yapıldığı gibi bir optimizasyon çalışmasında kullanılmak üzere birinci dereceden bir cevap yüzeyi oluşturmak için de kullanılabilir. Bunun yanı sıra ikinci derece yüzeyler, burada elde edilen sonuçların dışında az sayıda ek deneylerin yapılmasıyla elde edilebilmektedir.

Şekil 2-1'de iki parametrelili bir tasarım gösterilmektedir. Burada köşeler, parametrelerin biri düşük, diğeri yüksek seviye olmak üzere iki farklı değeriyle oluşturulan kombinasyonlardan her birini göstermektedir. Örneğin (1) ile gösterilen noktada hem A parametresi hem de B parametresinin düşük seviyede olduğu tasarım noktasıdır. Bir parametrenin *ana etkisi*, parametrelerdeki değişimin cevapta yarattığı değişikliğin diğer parametrelerin farklı değerleri için ortalaması alınmasıyla bulunur. İki parametrelili tasarımdan yola çıkarak A'nın ana etkisi,

$$A = \frac{ab + a}{2n} - \frac{b + (1)}{2n} \quad (2.4)$$



Şekil 2.1: 2^2 Faktöriyel Tasarımı

ile hesaplanır. Burada n ile gösterilen deneylerin tekrar sayısıdır. Yapılacak deneyler sayısal olup, sonuçlar değişim göstermeyeceği için $n=1$ alınmalıdır. Sözü açılmışken bu çalışmada normalde gerçek bir deneyde kullanılması gerekli olan fakat bu çalışmada kullanmayacağımız uygulamaları şu şekilde sıralayabiliriz;

- Deneylerin birden fazla kez yapılması (replikasyon)
- Deneylerin karışık sırada yapılması (randomizasyon)
- Bloklama denilen, deneylerin farklı zamanlarda bölüm bölüm yapılması

Bu uygulamalardaki amaç deneysel hataları veya cevaptaki değişimi (varyansı) hesaba katmaktır. Bu konularla ilgili olarak Box ve diğ. (2005) ve Myers ve Montgomery (2002)'e başvurulabilir.

Ana etkiyi hesaplarken kolaylık sağlaması açısından Tablo 2.1'deki işaret sisteminden faydalanılabilir.

Tablo 2.1: Etki Hesabı İçin İşaret Tablosu

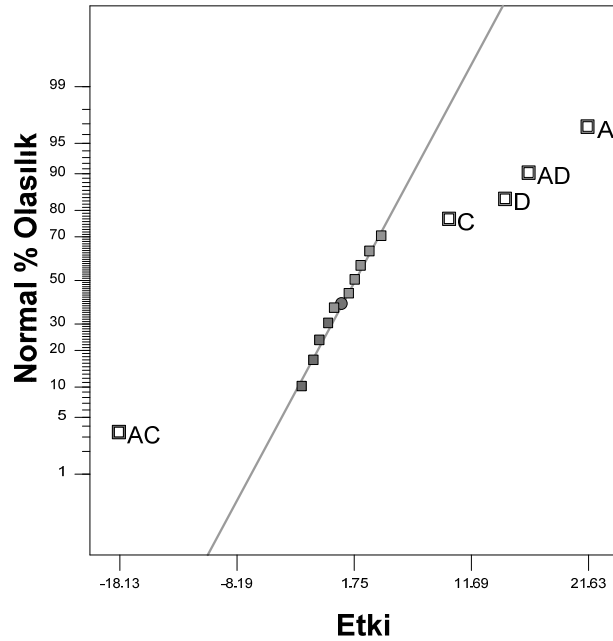
Kombinasyon	Faktöriyel Etki		
	A	B	AB
(1)	-	-	+
a	+	-	-
b	-	+	-
ab	+	+	+

Tabloda “kombinasyon” isimli kolonda parametrelerin deęiřik seviyelerinin oluřturduęu deneyler temsil edilmemektedir. Bu deneylerde her parametrenin seviyesi yksek veya dřk oluřuna gre tabloda sırasıyla + veya – ile iřaretlenir. Daha sonra etkileřim terimleri de hangi parametrelerden oluřuyorsa, o parametrelerin + ve – iřaretlerinin arpılmasıyla bulunur. rneęin b kombinasyonunun AB etkisi hesabındaki iřareti iin b satırındaki A ve B’nin iřaretleri olan – ve + arpılarak – iřareti bulunur.

Daha sonra etki hesabı iin ilgili parametrenin veya etkileřimin stnundaki iřaretler dikkate alınarak deney sonuları toplanır ve 2’ye blnr.

2.3.1 Modeli Oluřturacak Ana Etkilerin Belirlenmesi

Ana etkilerin seimini yapmak zere, hesaplanan etki deęerleri iin normal olasılık grafięi oluřturulur. řekil 2-2’de rnek bir grafik grlmektedir.



řekil 2.2: Etki Normal Olasılık Grafięi

Grafikte lineer bir eęilim gsteren noktaların katkısı modelde ihmal edilebilir dzeydedir. Dięer noktalar ise modelin ana etkileri olarak seilmelidirler. ANOVA hesabında ana etkiler modelin terimleri olarak, kalan noktalar da hata terimleri olarak kullanılır. Ana etkiler seildikten sonra hipotez testine geilir.

2.3.2 Model ve Model Terimleri İçin Hipotez Testi

Bir önceki bölümde işaretlerle yapılan (2'ye bölünmemiş) toplama *kontrast* adı verilir. Kontrast, ANOVA hesabında kullanılan *kareler toplamı* değerini hesaplarken kullanılır. Kareler toplamı,

$$KT_i = \frac{(\text{kontrast}_i)^2}{2^k} \quad (2.5)$$

ile hesaplanır. Denklemden k parametre sayısını, i de ilgili parametre veya etkileşimi göstermektedir. ANOVA'daki bir başka önemli değer *ortalama karesi* adı verilen ve her parametre ve etkileşimin kareler toplamının serbestlik derecelerine bölünmesi ile elde edilir. Serbestlik derecesi her parametre için 1'dir.

Toplam kareler toplamı ise,

$$KT_T = \mathbf{y}'\mathbf{y} - \frac{(\sum_{i=1}^n y_i)^2}{n} \quad (2.6)$$

olarak bulunur. Burada n deney sayısıdır.

Hata veya *artık*, gerçek sonuçla oluşturulan modelin verdiği cevap değerinin arasındaki farktır. Artık serbestlik derecesi (deney sayısı – parametre sayısı) dır. Artık kareler toplamı da toplam kareler toplamından bütün parametrelerin kareler toplamının toplamı çıkarılarak bulunur.

ANOVA'da F-testi için gerekli F değeri model için,

$$F_0 = \frac{MS_R}{MS_E} \quad (2.7)$$

ile hesaplanır. MS_R model ortalama karesi, MS_E de artık ortalama karesidir. Eğer F_0 değeri $F_{\alpha, k, n-k-1}$ (α :güven aralığı, n:deney sayısı, k:modeldeki terim sayısı) değerinden büyükse *sıfır hipotez* (H_0) denilen ve en az bir parametrenin bile cevap üzerinde belirleyici rol oynamadığı tezi reddedilmiş olur.

Model için yapıldığı gibi, parametreler ve parametreler arası ilişki için de hipotez testi yapılabilir. Buradan yola çıkarak modele yaptıkları katkılara göre regresyon modeline terim olarak dâhil edilip edilmeyeceğine karar verilebilir. Kısmi F-testi denilen bu yöntemde F_0 istatistik değerini

$$F_0 = \frac{KT_R(\beta_1 | \beta_2)/r}{MS_E} \quad (2.8)$$

ifadesi verir. MS_E model f-testi'ndeki gibi tam modelin artık ortalama karesidir. r değeri incelenen parametre veya parametre grubunun (tek bir parametre yerine birden fazla parametre veya ilişkinin katkısı da bu yolla test edilebilir) eleman sayısıdır. Örneğin bu değer tek bir parametre için hesaplanacaksa $r=1$ dir. KT_R ise *ekstra kareler toplamı* denilen ve hipotez testi yapılacak parametrelerin modele eklenmesi ile kareler toplamında ortaya çıkan artıştır. Ekstra kareler toplamı,

$$SS_R(\beta_1 | \beta_2) = SS_R(\beta) - SS_R(\beta_2) \quad (2.9)$$

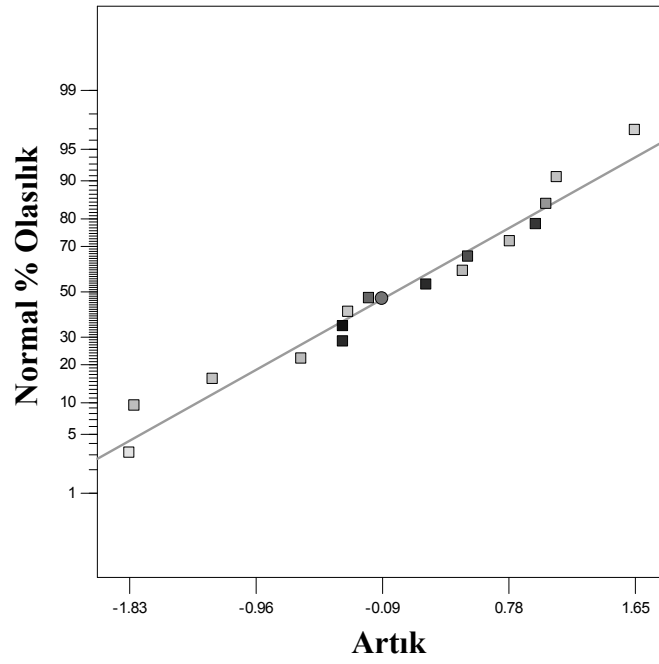
ile bulunur. Burada $SS_R(\beta)$ tam modelin, $SS_R(\beta_2)$ de β_1 'i (hipotez testi yapılan parametre veya parametreler) içermeyen modelin artık kareler toplamıdır.

F-testi, elde edilen F_0 değerinin $F_{\alpha, r, n-p}$ (α :güven aralığı, n :deney sayısı, p :modeldeki terim sayısı) değeriyle karşılaştırılmasıyla yapılır. Eğer F_0 değeri $F_{\alpha, r, n-p}$ 'den büyük ise eklenen parametrelerin modele katkısının olmadığını savunan H_0 hipotezi reddedilir.

2.3.3 Artık Analizi

Tahmin edilen cevap değeri ile gerçek değer arasındaki fark olan artık, oluşturulan modelin yeterliliğini test etmek için kullanılabilir.

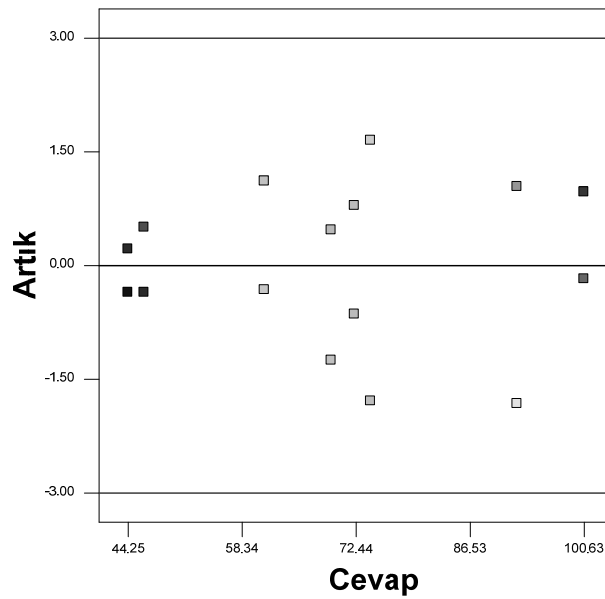
Normal olasılık grafiği, artıkların normal dağılımda olup olmadığının kontrolü için kullanılır. Şekil 2-3'de artık için normal olasılık grafiği görülmektedir. Artıkların grafikte yaklaşık olarak lineer bir dağılımda olması, bize normal dağılım varsayımının doğru olduğunu gösterir.



Şekil 2.3: Artık Normal Olasılık Grafiği

Bu grafikte lineer olmayan bir dağılım gözlemlenirse bir sonraki kısımda bahsedilen değişken dönüşümünün kullanılması gerekmektedir.

Bir diğer sıklıkla kullanılan kontrol yöntemi, tahmin edilen cevap değerlerine karşı artık değerlerin grafik yolla incelenmesidir. Şekil 2-4'te bir örneği gösterilen bu grafikte artıkların tamamen rasgele dağılması gerekmektedir.



Şekil 2.4: Cevap - Artık Grafiği

Rasgele olmayıp, belli bir motif gösteren durumlarda normal olasılık grafiğinde olduğu gibi değişken dönüşümü ile durumun kararlı bir hale sokulması sağlanabilir.

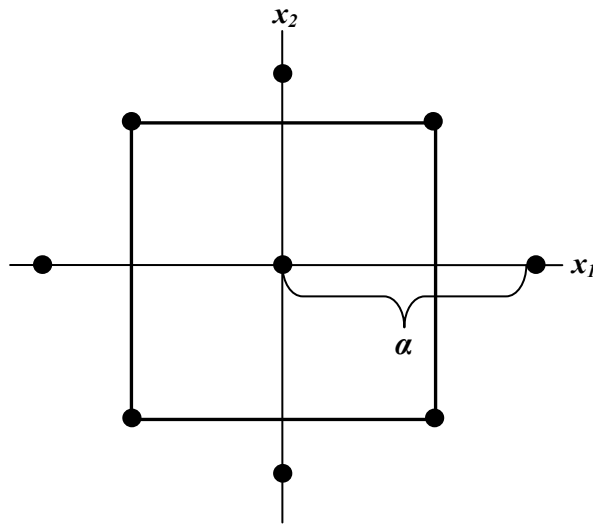
2.3.4 Cevap Değişken Dönüşümü

Modelin gerçek değerlere daha uyumlu olmasını sağlamak için kullanılan yöntemlerden biridir. Cevap değerlerinin en büyük değerinin en küçük değerine oranının 10'dan fazla olması durumunda dönüşüm uygulamak gerekmektedir.

Dönüşüm, cevap değişkenlerinin bir fonksiyon süzgecinden geçirilip, regresyonun elde edilen değerle yapılmasından ibarettir. Genellikle kuvvet fonksiyonu veya logaritma fonksiyonu kullanılmaktadır.

2.4 Merkezi Kompozit Tasarımı

Cevap yüzey yönteminde, öncelikle 1. dereceden bir model oluşturma çalışması yapılır. Bu çalışmada parametreler ve onların etkileşimini temsil eden etkileşim terimlerinden oluşan bir model oluşturulur. Oluşturulan modelde eğer etkileşim terimlerinin etkisinin az olduğu gözleniyorsa, parametrelerin 2. dereceden etkilerinin de az olacağı söylenebilir. Fakat bu etki fazla ise eldeki mevcut tasarım ilave deneylerle genişletilip, ikinci dereceden bir cevap yüzeyi elde etme yoluna gidilmelidir.



Şekil 2.5: İki Parametrelili Merkezi Kompozit Tasarımı

2. dereceden yüzeyler elde etmek için popüler bir yöntem olan merkezi kompozit tasarımı, mevcut 2^k tasarımına aksenel noktaların eklenmesi ve regresyon yapılmasına dayanır. Grafiksel olarak Şekil 2-5'te $k=2$ olan bir merkezi kompozit tasarımı görülmektedir.

Görüldüğü gibi köşeler dışında, bir adet merkezde ve her parametre için parametrenin $\pm\alpha$, diğer parametrenin (2'den fazla olduğunda, parametrelerin) 0 değerini aldığı deneyler tasarıma eklenir. α değeri parametre sayısına bağlı olarak

$$\alpha^2 = \sqrt{n_c} \quad (2.10)$$

ile belirlenir. Burada n_c , tam faktöriyel tasarımı için parametre sayısıdır. 2.10 denklemi ile bulunan α değeri, *döndürülebilirlik* denilen, tasarım uzayında merkez noktaya eşit mesafede iki tasarım noktasının tahmini cevap değerinin eşit derecede iyi olmasını sağlamaktadır. n_c değeri kesirsel tasarımlarda (bir sonraki bölümde tarif edilecek) ise parametre sayısı değil tam faktöriyel tasarımı gibi düşünülüp deney sayısına tekabül eden parametre sayısı kullanılmalıdır.

2.5 Kesirsel Faktöriyel Tasarımı

Bir deney tasarımında yapılması gereken deney sayısının parametre sayısına bağlı olduğu önceki bölümlerde bahsedilmişti. Deney sayısı, parametre sayısının üstel bir fonksiyonu olduğundan (2^k), belli sayıdan fazla parametre için deney yapılması fizibil değildir. En azından tam faktöriyel tasarım yapılması yerine, yüksek mertebeden ilişkilerin ihmal edilmesi pahasına kesirsel bir faktöriyel tasarımı uygulanabilmektedir.

2.5.1 Yarım 2^k Tasarımı

Bu tasarımda tam bir deney tasarımının yarısı kadar deney yapılır. Parametrelerin ana etkisi kendi aralarında bağımsız olarak ifade edilir. Fakat bu etkiler ikili veya daha fazla sayıda parametre arası ilişki ile kombine olarak göz önünde bulundurulur. Tablo 2-2'de bu tip bir tasarımdaki etki değerlerinin hesabı için hazırlanmış işaret tablosu yer almaktadır.

Tablo 2.2: Yarım 2k Tasarım İşaret Tablosu

	Faktöriyel Etki							
Kombinasyon	I	A	B	C	AB	AC	BC	ABC
a	+	+	-	-	-	-	+	+
b	+	-	+	-	-	+	-	+
c	+	-	-	+	+	-	-	+
abc	+	+	+	+	+	+	+	+
ab	+	+	+	-	+	-	-	-
ac	+	+	-	+	-	+	-	-
bc	+	-	+	+	-	-	+	-
(1)	+	-	-	-	+	+	+	-

Tasarım tabloda kalın çizgiyle belirtilmiş yerden yukarıda veya aşağıda kalan deneylerle yapılmalıdır. Bu tasarımda *belirleyici ilişki* adı verilen ve

$$I = ABC$$

ile gösterilen ifade, seçilen alt kümede kolon boyunca birim terim I gibi + işareti alan etkileşim terimlerini göstermektedir. Hesaba beraber olarak katılacak terimleri bulmak için terimler ABC (bu örnek için) ile çarpılıp gerekli sadeleştirmenin yapılması gerekmektedir. Örneğin C teriminin hangi terimle hesaba katılacağını bulmak için

$$C.ABC = ABC^2 = AB$$

Dolayısıyla yapılacak olan yarım tasarımda C ve AB kombine olarak düşünülüp hesaba katılacaktır. Burada bir terimin kendisi ile çarpımının birim terim I olduğu kullanılmıştır.

Yukarıda bahsedilen farklılıklar dışında tam faktöriyel bir dizayn için uygulanan metodoloji burada da aynen kullanılacaktır.

2.6 Ek Bilgi

Deney tasarımı ve cevap yüzey metodolojisi gerçek deneylerde hata analizi ve sonuç değerlerinde gözlenen değişimi de hesaba katar. Daha önce de bahsedilen bu konunun dışında sıklıkla kullanılan Box-Behnken (1960) tasarımı, karışım modellemesi ve modelleme sonrası uygulamak üzere optimizasyon prosedürleri gibi

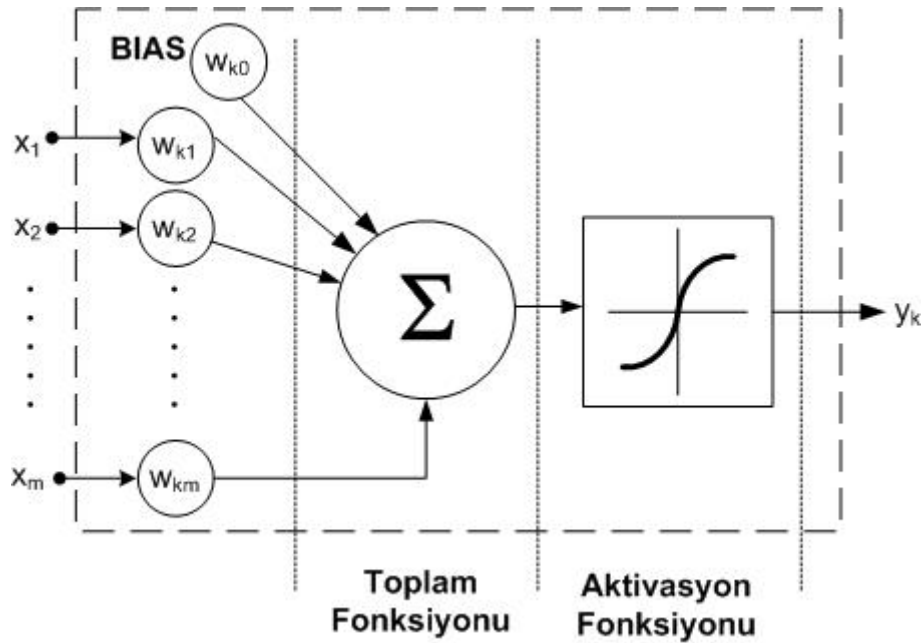
bu alıřmada yer almayan konularla ilgili olarak ve daha detaylı bilgi almak iin Box ve diğ. (2005) ve Myers ve Montgomery (2002)'e bařvurulması tavsiye edilmektedir.

3. YAPAY SİNİR AĞLARI

Yapay sinir ağı, canlılardaki gerçek sinir ağlarından hareketle bulunan bilgi işleme sistemlerine verilen genel addır. Bu sistemler kontrolden örüntü tanımaya, modellemekten tıbbi teşhise ve finansal tahmin konularına kadar çok geniş bir yelpaze de kullanım imkânı bulmaktadır.

Bu kadar çeşitli alanda tercih edilmesini sağlayan etkenlerden biri olan lineer olmaması, özellikle lineer olmayan fiziksel mekanizmaların modellenmesinde işe yaramaktadır. Bir diğer özelliği ise adaptif bir yapıda olmasıdır. Özellikle dinamik sistemlerin modellenmesinde, verilerin değişimine cevap vererek sil baştan bir model kurulumuna gerek duyulmamaktadır.

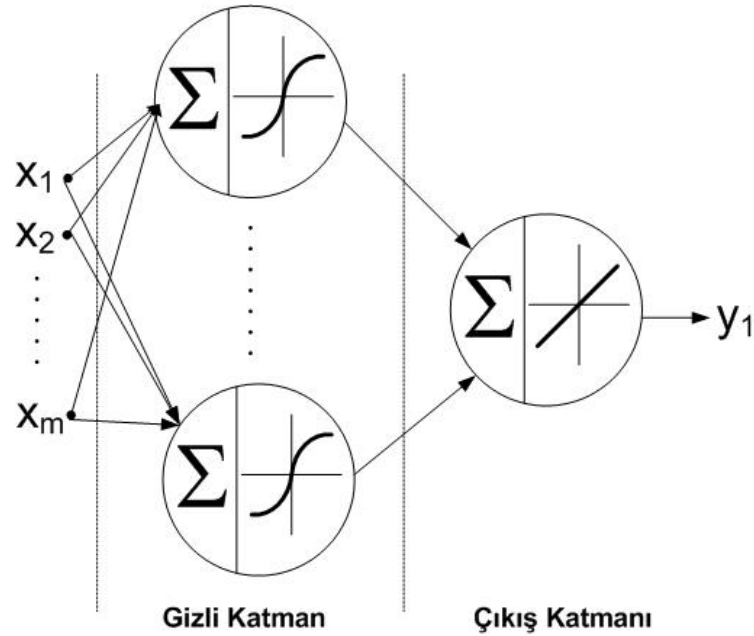
Ayrıca bu sistemlerin yazılımsal değil donanımsal (VLSI teknolojisi) olarak hayata geçirilebilmesi de çok önemlidir. Bu bağlamda dağılımsal bir yapıya sahip olmasından ötürü hatayı tolere edebilme özelliği ve paralellik arz eden bir topolojiye



Şekil 3.1: Nöron Modeli

sahip olması da hesaplama hızı bakımından bu yöntemi alternatiflerine nazaran öne çıkarmaktadır.

Bir yapay sinir ağı, gelen giriş sinyallerinin belirli katsayılarla ağırlıklandırılmış toplamının bir eşik fonksiyonundan geçirildiği, *nöron* adı verilen birim elemanların birbirlerine bir ağ yapısıyla bağlanmasıyla oluşturulmaktadır. Şekil 3-1’de örnek bir nöron modeli gösterilmiştir. Görüldüğü gibi her giriş değeri w ile gösterilen ağırlık katsayıları ile çarpılarak toplam fonksiyonuna gönderilmektedir. “Bias” adı verilen ve giriş değerlerinden bağımsız sabit bir sayı da bu toplamda yer almaktadır. Bias için genellikle 1 değeri kullanılır. Bu ağırlıklandırılmış toplam “*aktivasyon fonksiyonu*” adı verilen ve bir eşik fonksiyonu görevi üstlenen, genelde şekildeki gibi lineer olmayan bir fonksiyona tabi tutulur.



Şekil 3.2: İleri Beslemeli Ağ

Şekil 3-2’de bir ileri beslemeli yapay sinir ağı modeli görülmektedir. Modellemede sıkça tercih edilen bu ağ tipinde genelde iki adet “*katman*” adı verilen nöron grubu yer alır. Reel sayılar uzayında tanımlı neredeyse her fonksiyon bir *gizli katman* ve bir *çıkış katmanı* ile modellenenbilmektedir. Verinin ileri doğru yayıldığı bu ağda öncelikle her giriş değeri ilk gizli katmanda bulunan her nörona iletilir. Daha sonra her nöronun çıkış değeri, bulunduğu katmanı takip eden katmana giriş değeri olur ve bu şekilde bilgi son katmana kadar yayılır. Son katmanda fonksiyonun çıkış değişkeni kadar nöron yer almalıdır.

Örnek modelde de görüldüğü gibi gizli katmanda lineer olmayan bir aktivasyon fonksiyonu (örneğin *sigmoid* fonksiyonu), çıkış katmanında ise lineer bir fonksiyon kullanılır. Lineer olmayan fonksiyon tercihi, bu fonksiyonların geri yayılım algoritmasında gerekli olan “türevlerinin alınabiliyor olması” özelliğinden kaynaklanmaktadır.

3.1 Öğrenme

Temel olarak, ağa verilen giriş veya giriş-çıkış değerlerinden ağdaki katsayıların belirlenmesi ve sonra ağa verilen girişlerden çıkış değerlerin tahmin edilmesi hedeflenmektedir. Burada yalnızca giriş değerlerinin var olması ile “*eğitimsiz öğrenme*” denilen ve verilerin birbirlerine yakınlığına göre sınıflandırılması amaçlı yöntemler kastedilmektedir. Bu tezde kullanılan “*eğitilmiş öğrenme*” adı verilen ve gerçek çıkış değeri ile ağın tahmin çıkış değeri arasındaki farkın, yani hatanın, ağa geri-bildirimiyle ağırlıkların hatayı minimize edecek şekilde güncellenmesine dayanan öğrenme yöntemi kullanılmaktadır.

Diğer alanlardaki kullanımını bir kenara bırakarak bu noktadan sonra bu tezde ihtiyacımız olan fiziksel bir olayın modellenmesindeki kullanımı üzerinde duracağız. Bu bağlamda öncelikle bu sistemin iki önemli fazını inceleyelim; modelleme kabiliyeti ve modellemenin nasıl yapılacağı.

3.2 Evrensel Yaklaşım ve Kolmogorov Teoremi

Stone-Weierstrass teoreminden yola çıkılarak elde edilen *evrensel yaklaşım teoremi*, yapay sinir ağlarının fonksiyon yaklaşımı açısından uygunluğu Hornik ve diğ. (1989) tarafından gösterilmiştir. Bu teoreme göre $[0,1]$ aralığında tanımlı sürekli ve çok boyutlu bir f fonksiyonu istenilen yakınlıkla

$$F(x_1, \dots, x_m) = \sum_{i=1}^{m_1} \alpha_i \varphi \left(\sum_{j=1}^{m_0} \omega_{ij} x_j + b_i \right) \quad (3.1)$$

ile yaklaşık olarak ifade edilebilir. Burada α_i, b_i ve ω_{ij} reel sabitler olup φ ise sabit olmayan, sınırlı ve monoton artan sürekli bir fonksiyondur. φ fonksiyonu olarak genelde sigmoid adı verilen

$$\varphi(v) = \frac{1}{1 + e^{(-av)}} \quad (3.2)$$

fonksiyon kullanılır.

Weierstrass teoremi tek değişkenli fonksiyonlar için yaklaşımı garantiler iken Kolmogorov (1957) teoremi ise çok değişkenli sürekli fonksiyonların, tek değişkenli sürekli fonksiyonların toplama işlemi ile süperpozisyonundan elde edilebileceğini göstermektedir.

Buradan yola çıkarak Şekil 3-2’de görülen iki katmanlı ağın gizli katmanı denklem (3.1) deki forma uymakta ve $[0,1]$ aralığında sürekli bir fonksiyonu modelleyebilmektedir. Çıkış katmanında bulunan lineer aktivasyon fonksiyonu da çıkış değişkeninin reel sayılar kümesinden herhangi bir değeri alabilmesini sağlamaktadır.

Sonuç olarak Weierstrass ve Kolmogorov teoremleri bize bir yapay sinir ağının sürekli bir fonksiyonu modelleyebildiğini göstermektedir. Buradan hareketle yapay sinir ağlarının modelleme yapabilmesindeki en önemli nokta olan ve öğrenme adı verilen, denklem (3.1)’deki reel sayıların en uygun şekilde tespit edilmesi konusunda bir sonraki bölümde bilgiler verilecektir.

3.3 Geri Yayınım Algoritması

Modellemeyi, elde bulunan veya üretilen (bu çalışmada olduğu gibi) giriş-çıkış çiftlerinden oluşan bir veri kümesinden yola çıkarak, fiziksel bir olayın davranışının tahmin edilebilmesi olarak özetleyebiliriz. Yapay sinir ağları uygun katsayıların seçilmesi durumunda verilen giriş değerleri için olası çıkış değerlerini istenilen hassasiyette verebilmektedir.

Bunun için ağın öğrenme denilen ve mevcut katsayılarla elde edilen çıkış değerinin olması gereken çıkış değeri ile karşılaştırılması ve neticede katsayılarda uygun güncellemelerin yapılması gerekmektedir. Bu güncellemeler hatanın minimize edilmesi hedefinden yola çıkılarak yapılmaktadır.

İlginçtir ki yapay sinir ağları bu tez çalışmasında optimizasyona yardımcı olmak üzere kullanılacaktır. Fakat yapay sinir ağları da çalışması için bir optimizasyon probleminin çözümüne ihtiyaç duymaktadır. Böyle iç içe geçmiş ilişkiler, problemler

farklı farklı gözükseler de temelde aynı felsefeden türemelerinden ötürü, daha birçok yerde karşımıza çıkacaktır.

Tabii ki genetik algoritmalar da yapay sinir ağlarının optimizasyonu için düşünülebilir. Ancak problem nispeten daha basit olduğu için genellikle konvansiyonel yöntemler hem daha hızlı hem de daha doğru sonuçlar üretebilmektedir.

3.3.1 En-Dik İniş Algoritması

Öncelikle en basit yöntem olarak hata fonksiyonunun gradyanını kullanarak optimizasyonu gerçekleştiren en-dik iniş algoritmasından bahsedelim.

En dik iniş algoritması ile fonksiyonun tanımlı olduğu herhangi bir noktadan başlanıp, fonksiyonun gradyanı yani en hızlı artışın olduğu yön kullanılarak maksimum ya da minimum noktaya ulaşılır. Kısaca,

$$x_{i+1} = x_i + \mu \nabla f(x_i) \quad (3.3)$$

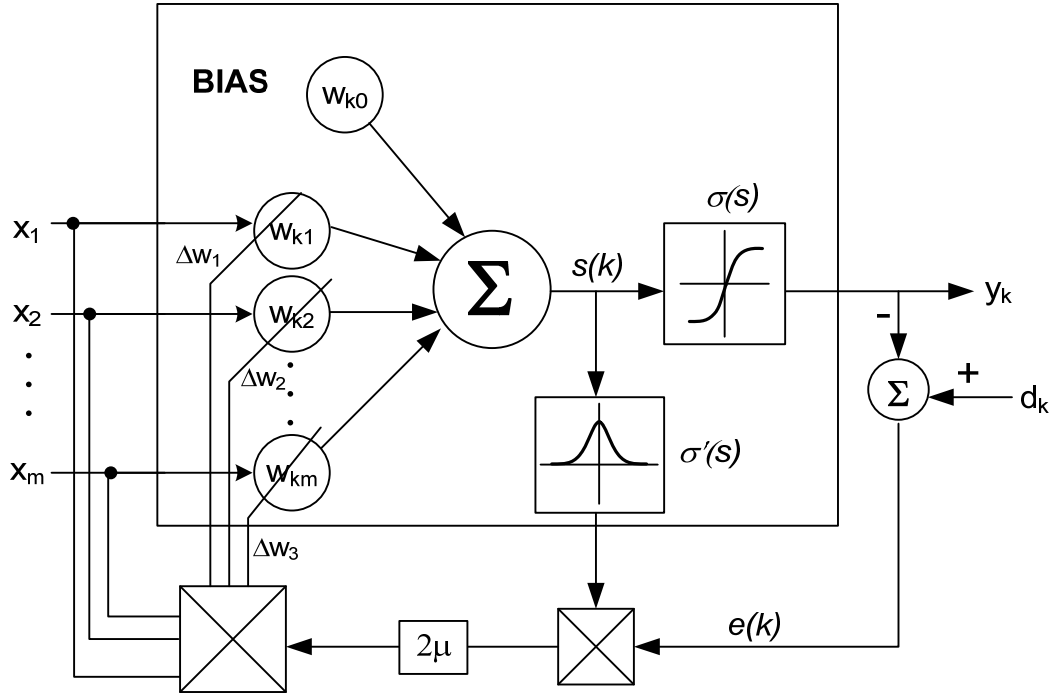
şeklinde gösterilebilir. Burada μ yeterince küçük reel bir sayıdır. Bu sayının büyük olması arama hızını arttırabilir fakat hassasiyeti azaltacağından ekstremum noktasını kaçırmak olasıdır.

Yapay sinir ağlarındaki kullanımına gelince, öncelikle minimizasyonu yapılmak üzere hata fonksiyonu seçilir. Bu fonksiyon sıklıkla hata veya hataların kareler toplamıdır. Ağırlık adı verilen nörondaki katsayıların değiştirilmesiyle hata minimize edileceğinden gradyan ağırlık değerlerine göre alınır.

Detaya girmeden nihai olarak elde edilen bir nöron için ağırlık güncelleme algoritması

$$\mathbf{w}_a(k+1) = \mathbf{w}_a(k) + 2\mu e(k) \sigma'(s(k)) \mathbf{x}_a(k) \quad (3.4)$$

şeklinde elde edilir(Gupta ve diğ., 2003). Burada $\mathbf{w}$ ağırlık vektörü, $\mathbf{x}$ giriş değerleri vektörü, μ denklem 3.3'teki sabit, σ aktivasyon fonksiyonu, s toplam fonksiyonu ve k da iterasyon numarasıdır. Şekil 3-3'te bu geriye hata beslemeli nöronun şematik versiyonu da görülmektedir.



Şekil 3.3: Hata geri besleme

Denklem (3.4) bir adet nöron için öğrenme formülüdür. Fakat pratikte çok katmanlı ağlar kullanılmaktadır. Çok katmanlı ağlar için öğrenmede hataların, nöronların ağırlık katsayılarının büyüklüklerine göre nöronlara yansıtılması yaklaşımı ile yapılmaktadır. *Geri yayılım algoritması* ifadesi ile çok katmanlı hal için uygulanan öğrenme kastedilmektedir.

Çok katmanlı hal için geri yayılım yönteminin matematiksel çıkarımı üzerinde durulmayacak olup, aşağıda adım adım izlenmesi gereken prosedür verilmektedir:

1. Ağdaki her nörona rasgele seçilmiş küçük reel sayılar atanır ve pozitif bir μ değeri belirlenir,
2. Bir giriş-çıkış değeri ağa uygulanır (İleri yayılım),
3. Hata değeri ve çıkış katmanı için hata sinyali aşağıdaki gibi hesaplanır:

$$E = \frac{1}{2} \sum (d - y)^2 \quad (3.5)$$

$$\delta = (d - y) \sigma'(s_{\text{çıkış}}) \quad (3.6)$$

burada d gerçek değer, y ağ çıkış değeri, δ hata sinyalidir.

4. (Hata geri yayını) Gizli katmanlardaki nöronlara hata geri yayını uygulanır. Bunun için hata sinyali değeri,

$$\delta_n = \sigma'(s_n) \sum w_{mn} \delta_m \quad (3.7)$$

ile hesaplanır. n indisi hata sinyali hesaplanan nöronu, m indisi de bir ileri katmanı temsil etmektedir. Toplam sembolü, bir ileri katmandaki bias hariç her nöronun (bias'a önceki katmandan sinyal gelmediği için hata sinyali yoktur) ağırlık ve hata sinyali değerinin çarpımlarının toplanmasını ifade etmektedir. s_n değeri nöronun aktivasyon fonksiyonuna uygulanan toplam değerini göstermektedir.

5. Son olarak ağırlık değerlerinin güncellenmesi için,

$$\Delta w_{mn} = \mu \delta_m y_n \quad (3.8)$$

ifadesi kullanılır. Burada w_{mn} değeri n nöronundan m nöronuna sinyal geçişinde uygulanan katsayıyı temsil etmektedir. y değeri n nöronu çıkış değeri, δ değeri de m nöronunun önceki adımda hesaplanan hata sinyali değeridir.

6. 2. adımdan 5. adıma kadar anlatılanlar her giriş-çıkış veri çifti için uygulanmalıdır. Bütün verilerin bu şekilde ağa uygulanmasına bir *devir* adı verilir.
7. Bir devir sonunda elde edilen hata değeri kabul edilebilir sınırlar içinde değilse bir sonraki devire geçilir, yani bütün veri çiftleri bir kez daha uygulanır. Eğer hata yeterince küçükse uygulama sonlandırılır (Lin, 1996)

Geri yayını, en-dik iniş yöntemi ile çok iyi performans gösteremez. Bunun sebebi hata fonksiyonunun sadece birinci türevlerinin değerlendirilmesi ve dolayısıyla lokal yakınsamanın daha zor gerçekleşmesidir. Önümüzdeki bölümde orta ölçekte bir hata minimizasyonunda kullanılabilen ve yakınsama hızı daha iyi olan bir yöntemden bahsedilecektir.

3.3.2 Levenberg-Marquardt Algoritması

Ekstremum civarında yakınsama anlamında en-dik iniş yöntemine nazaran daha iyi bir performans gösteren Gauss-Newton algoritması ne yazık ki daha geniş aralıklarda daha başarısızdır.

Levenberg-Marquardt algoritması (Levenberg, 1944 ve Marquardt, 1963) her iki yöntemin olumlu taraflarını kullanan kararlı bir iteratif optimizasyon algoritmasıdır. Hata fonksiyonunun Taylor serisine açılımını (ikinci derece terimlere kadar)

$$E(\mathbf{w} + \Delta \mathbf{w}) = E(\mathbf{w}) + (\mathbf{J} \mathbf{E})^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T (\mathbf{J}^T \mathbf{J}) \Delta \mathbf{w} \quad (3.9)$$

şeklinde gösterebiliriz. Burada $\mathbf{w}$ ağırlık vektörünü, $\mathbf{E}$ hata değerleri vektörünü, $E(\mathbf{w})$ hataların karelerinin toplamı olan fonksiyonu ya da diğer bir deyişle $\mathbf{E}$ 'nin normunu, $\mathbf{J}$ hata vektörü $\mathbf{E}$ 'nin Jacobian'ını, $\mathbf{J} \mathbf{E}$ çarpımı E fonksiyonunun gradyanını, $\Delta \mathbf{w}$ ağırlık vektöründeki değişimi ve $\mathbf{J}^T \mathbf{J}$ de $\mathbf{E}$ vektörünün Hessian'ını göstermektedir. $\mathbf{J}^T \mathbf{J}$ aslında gerçek Hessian matrisi olmayıp bir yaklaşımdır. Bu yolla sadece birinci türev değerleri kullanılarak ikinci türev bilgisi yaklaşık olarak kullanılabilmekte ve hesaplama sırasında hafızada daha az bilgiyle çalışabilmeyi sağlamaktadır.

Bu hata fonksiyonunun $\Delta \mathbf{w}$ 'ya göre gradyanı sıfıra eşitlenirse,

$$(\mathbf{J}^T \mathbf{J}) \Delta \mathbf{w} = -\mathbf{J} \mathbf{E} \quad (3.10)$$

şeklinde bir denklem elde edilir. Bu denklemin çözümü ile yapılan ağırlık güncellemesi yöntemi Gauss-Newton algoritması olarak adlandırılır.

Levenberg-Marquardt algoritması denklemin sol tarafına bir ekleme yaparak,

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}) \Delta \mathbf{w} = -\mathbf{J} \mathbf{E} \quad (3.11)$$

haline getirir. Burada λ değeri bir reel sayıyı, $\mathbf{I}$ ise birim matrisi göstermektedir. Yöntemde, başlangıç aşamasında λ değeri büyük alınır. $\lambda \mathbf{I}$ çarpımının baskın olması ile denklem en dik iniş algoritması gibi çalışır. Hata değerinde hızlı düşüş görüldüğünde λ değeri azaltılarak Gauss-Newton algoritmasına yaklaşılr. Tam tersine, yavaş bir azalma gözleniyorsa λ değeri arttırılmalıdır.

Yöntemin yapay sinir ağlarına uygulamasını da Hagan ve Menhaj (1994) gerçekleştirmiştir.

3.4 Giriş Çıkış Değerleri Normalizasyonu

Giriş ve çıkış değerleri, öğrenmenin daha hızlı ve efektif yürütmesi adına normalleştirmeye ihtiyaç duyar.

Giriş değerleri genelde en küçük değeri -1 ve en büyük değeri +1 olacak şekilde $[-1,+1]$ aralığına ölçeklenirler. Bu sayede giriş değişkenlerinin kovaryanslarının yakın seviyelerde tutulması ile nöron ağırlıkları birbirine yakın hızlarda öğrenme gerçekleştirir (LeCun, 1993).

Çıkış değerlerini ele alırsak, minimum ve maksimum değer arasında yüksek(örneğin 10 kattan fazla) bir fark varsa öğrenme sırasında hesaplanacak hata değerleri de o mertebede değişiklik gösterecek ve ağırlık güncellemelerinde büyük salınımlar yapacaktır. Bu durumu iyileştirmek adına çıkış değerleri bir logaritma dönüşümüne sokulabilir. Böylelikle çıkış değerleri arasındaki mertebeye farkları ortadan kaldırılmış olur.

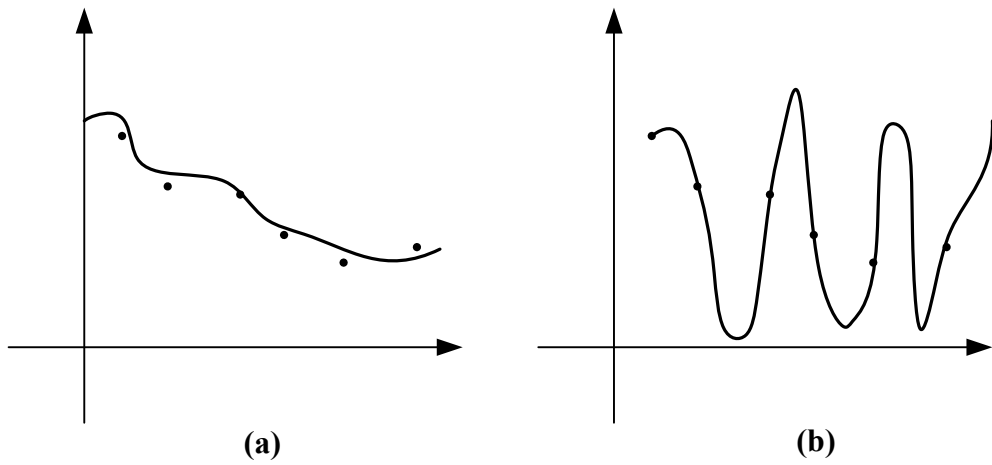
3.5 Genelleştirme

Önceki bölümlerde yapay sinir ağlarının, verilen giriş-çıkış değer çiftleri kullanılarak bu değerlere en yakın tahminleri yapmak üzere nasıl eğitildiklerinden bahsedildi. Oluşan hata değerinin minimizasyonu (eğer global minimum bulunabiliyorsa) bize, eğitim için kullanılan değerlerin giriş değerlerini ağa gönderdiğimizde elde edilecek çıkış değerinin, gerçek çıkış değerine olabilecek en yakın değer olacağını garantiler. Fakat asıl amaç, farklı giriş değerleri gönderildiğinde de gerçek değerlere çok yakın değerler verecek şekilde ağı eğitebilmektir.

Ağın böyle farklı değerler için de iyi sonuçlar verebilme yeteneğine *genelleştirme* adı verilmektedir. Bir ağın genelleştirme yeteneğini belirleyen faktörleri şöyle sıralayabiliriz:

1. Elde bulunan öğrenme için kullanılacak veri miktarı: Olayla ilgili ne kadar çok bilgi varsa o kadar iyi modelleme yapılabileceği anlamına gelir. Fakat belli bir miktardan sonra ağın yeterli büyüklükte olmaması bilgiye doyumu sağlayacak ve fazla bilgi öğrenmeye önemli katkılar sağlayamayacaktır.

2. Ağın mimarisi: Mimari denilince bir ağdaki katman sayısı, katmanlardaki nöron sayısı, nöronların birbirlerine olan bağlantı durumları vs. anlaşılır. Temelde bir ağ ne kadar gereksiz derecede karmaşık ve büyük bir mimariye sahipse *aşırı-uyum* adı verilen problemten etkilenecektir. Aşırı-uyum, bir ağın eğitim değerlerine çok yakın değerler döndürüp genelleştirmenin çok zayıf olması anlamına gelmektedir. Şekil 3-4'te tek değişkenli bir fonksiyon için bir aşırı-uyum durumu örneklenmektedir. Şekil 3-4 (a)'da noktalara tam uyumlu olmayan fakat genelde çok hatalı olmayan bir yol izleyen, Şekil 3-4 (b)'de ise noktalara tam uyum sağlayan fakat diğer bölgelerde normalde



Şekil 3.4: Aşırı-Uyum Problemi

beklenmeyecek aşırı salınımlar yapan ve aşırı-uyum sorunu olan yaklaşım fonksiyonu gösterilmiştir.

3. Modellenecek problemin karmaşıklığı: Problemin karmaşıklığının artmasıyla ağın genelleme başarısı (aynı sayıdaki eğitim verisi için) düşer(Haykin, 1999).

Bir yapay sinir ağında genelleştirme kabiliyetinin ekstra bir maliyet getirmeden artırılabilmesi ancak yukarıda belirtilen 2. maddeyle ilgili bir çözüm bulunmasıyla gerçekleştirilebilir. Bu konuyla ilgili olarak şimdi iki farklı yaklaşımdan bahsedilecektir.

3.5.1 Erken Durdurma

Öğrenme sırasında devir sayısının artmasıyla hata değeri gittikçe düşer. Verilerin böyle tekrar tekrar beslenmesiyle ağ en sonunda belli bir hata değerine yakınsar. Bu

prosedürde aslında ağ ilk aşamada değeri girilmeyen farklı durumlar için de öğrenme gerçekleştirir. Fakat gereğinden fazla yapılan bu öğrenme işi eldeki veriye daha sıkı bir uyum sağlayıp, diğer bölgelerde ideal konumundan uzaklaşır (Haykin, 1999).

Erken durdurma adı verilen yaklaşımla eldeki verilerin bir kısmı, öğrenme aktivitesi sürerken, genelleştirme kabiliyetindeki değişimi kontrol etmek üzere kullanılır. Öğrenmenin ilk zamanlarında hem eğitim kümesindeki hem de kontrol kümesindeki değerler için hata değerinde azalma gözlenir. Belli bir devir sayısından sonra kontrol kümesindeki hata değerindeki azalma durur ve hatta azalma yerini artışa bırakır. İşte tam bu noktada algoritma öğrenme işlemini sonlandırır ve genelleştirme kabiliyetinin azalmasına izin verilmemiş olur.

3.5.2 Ağ Budama

Genelleştirmeye yardımcı olmak üzere kullanılan bir diğer yöntem de “ağ budaması”dır. Bu yöntemde genelleştirme, ağın karmaşıklığının azaltılması ile sağlanmaktadır. Önceki bölümlerde bahsedilen, minimize edilmesi gereken hata kareleri toplamının yanına, bir de karmaşıklığı cezalandıran bir terim eklenir. Dolayısıyla minimize edilecek fonksiyon,

$$\mathbf{R} = \mathbf{E} + \lambda \mathbf{W} \quad (3.12)$$

olarak gösterilebilir. Burada $\mathbf{W}$ karmaşıklığı temsil eden bir fonksiyondur ve *ağırlık düşüşü* yaklaşımında

$$\mathbf{W} = \sum w_i^2 \quad (3.13)$$

şeklinde yani ağdaki bütün ağırlık katsayılarının karelerinin toplamı olarak alınmaktadır.

Ağırlık düşüşü yöntemi, nispeten daha büyük ve karmaşık ağ yapılarında cezalandırmayı kullanarak, modellemeye katkısı az olan ağırlıkların sıfıra yakın bir değer almasını sağlayarak alacakları rasgele değerlerle ağın verilere aşırı-uyum sağlamasını engellemektedir.

4. STİFNERLİ BİR PANELİN LİNEER BURKULMA MODELLEMESİ

Bu bölümde, önceki bölümlerde bahsedilen deney tasarımı ve yapay sinir ağları yöntemleri gerçek bir mühendislik problemine uygulanacaktır. Burada asıl amaçlanan yapay sinir ağlarının performansının, referans olarak alınan istatistiksel yönteme göre incelenmesidir.

Problemin tanımı ve uygulama aşamalarına geçmeden önce, izlenecek yolu özet olarak ifade etmek gerekirse, ilk olarak yapılması gereken eldeki problemin parametrelerinin belirlenmesidir. Bu aşamada probleme önemli etkileri olan parametrelerin gözden kaçması, modellemenin sağlıklı bir şekilde yapılamayacağı anlamına gelmektedir.

Daha sonra problem, simülasyonu yapılmak üzere modellenir. Model gerekli verileri üretmek üzere parametrik olarak hazırlanmalıdır. Mevcut çalışmada simülasyon için ANSYS (2005) adlı hazır bir sonlu elemanlar paketi kullanılmıştır. Harici olarak çalışan bir program gerekli giriş değerlerini alıp, her ayrı durum için sonlu elemanlar programına gönderilmek üzere bir programcık üretmekte ve simülasyon sonuçlarını bir dosyaya kaydetmektedir. Yani yazılan program aslında parametrik modeli temsil etmektedir.

Sonraki aşamalarda, iki yöntemin (yapay sinir ağları ve deney tasarımı) uygulamada farklılaştığı görülmektedir. Modellemede kullanılacak veriler deney tasarımında önceden belirli bir şekilde, yapay sinir ağlarında ise rasgele üretilmektedir. Daha sonra verilere göre uydurulacak modelin yapısı, deney tasarımında verilerden yola çıkılarak, yapay sinir ağlarında ise tahmin ile belirlenmektedir.

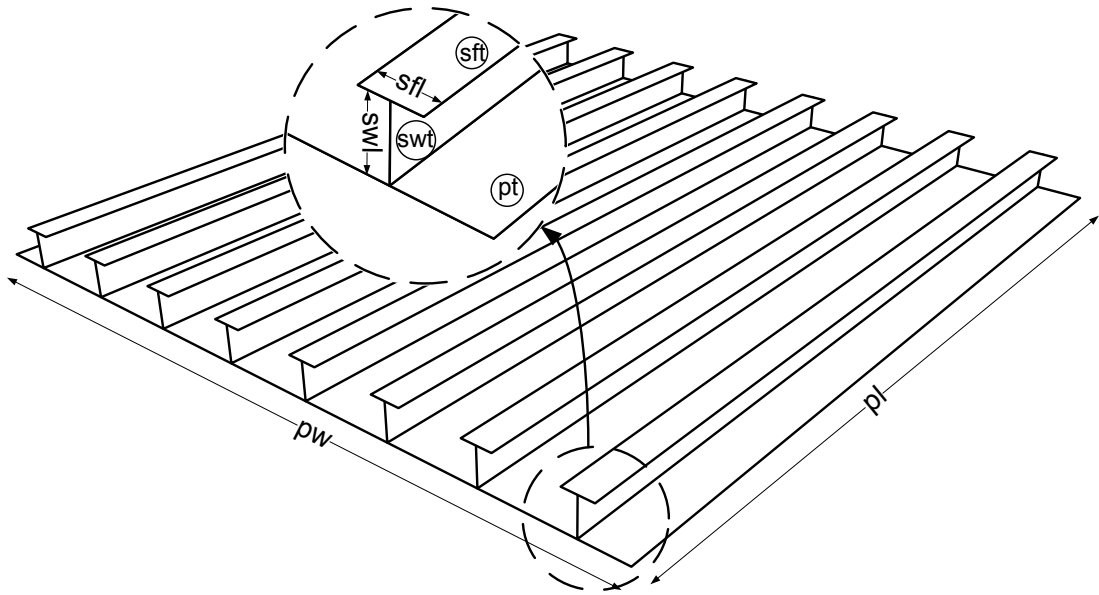
En sonunda da model veriye uydurulur ve sonuçlar rasgele üretilmiş farklı durumlar için kontrol edilir.

Şimdi problemin tanımına geçilecek olup, ardından sırasıyla deney tasarımı ve yapay sinir ağlarıyla modelleme yapılacak ve elde edilen sonuçlar karşılaştırmalı olarak incelenecektir.

4.1 Problemin Tanımı

Bu çalışmada daha önceki bölümlerde anlatılan yöntemlere gerçek bir uygulama olması açısından gemi yapılarında sıkça karşılaşılan, fakat kara ve kıyı yapılarında da karşımıza çıkan stifnerlenmiş bir panelin burkulma hadisesi modellenecektir.

Bir denge problemi olan *burkulma*, basma kuvvetine maruz kalan bir yapının, denge konumundan bir miktar saptırılmasıyla farklı bir denge konumuna geçmesi ve hatta yapının çökünceye kadar şekil değiştirmesi olarak tarif edilebilir. Diğer bir deyişle iki komşu denge durumunun farklı deformasyon modlarına karşılık gelmesidir(Gambhir, 2004). Yapının bu durumuna *kararsız denge* adı verilmiş olup, kuvvetin böyle kararsızlık yaratacak en düşük değerine de *burkulma yükü* veya *kritik yük* adı verilir. Bu uygulamada amaç, belirlenen parametrelerin verilen aralıkta aldığı herhangi bir değer kombinasyonu için yapının kritik yük değerinin tahmin edilebilmesidir.



Şekil 4.1: Stifnerli Panelin Geometrisi

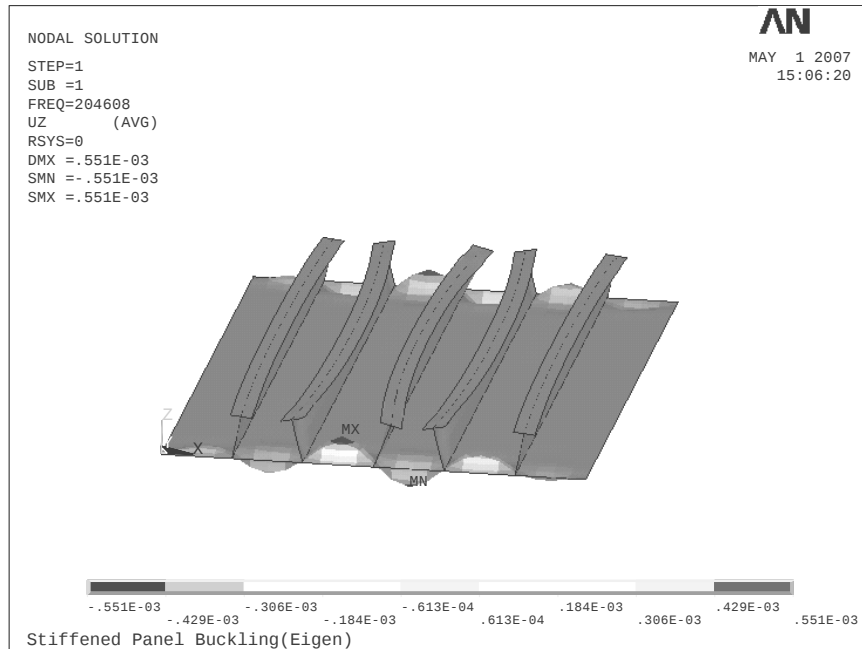
Modellenecek stifnerli panelin geometrisi Şekil 4-1’te görülmektedir. Görüldüğü gibi yapı, dikdörtgen şeklinde bir panele belirli sayıda T kesitli profillerin eklenmesiyle oluşturulmuştur. Bu örnek yapının homojen olarak çelikten üretildiği göz önünde bulundurularak, model için sadece geometrik parametreler seçilmiştir. Bu parametreler: panelin eni ve boyu, stifner sayısı, profil yükseklikleri ve alın genişlikleri ve panel, profil gövde ve alınlarının ayrı ayrı kalınlık değerleridir.

Seilen bu parametrelerin hangi aralıkta modelleneceėi ile ilgili bilgi de Tablo 4-1’de yer almaktadır.

Tablo 4.1: Dizayn Parametreleri ve Tanım Aralıkları

Parametre	İnceleme Aralığı	
	Min.	Maks.
Panel eni (pw)	2m	10m
Panel boyu (pl)	3m	15m
Panel kalınlık (pt)	3mm	10mm
Stifner sayısı (nos)	2	6
Stifner gövde yüksekliėi (swl)	20cm	40cm
Stifner gövde kalınlığı (swt)	2mm	5mm
Stifner alın geniřliėi (sfl)	10cm	20cm
Stifner alın kalınlığı (sft)	2mm	5mm

Sınır kořullarına gelince, panel 4 kenarından da basit mesnetli olarak kabul edilmiřtir. Uygulanacak yük de stifnerlerin doėrultusunda panele uygulanacaktır. Yukarıda tanımlanan dizayn simülasyona sokulmak üzere parametrik olarak hazırlandıktan sonra veri üretme aşamasına geçilir.



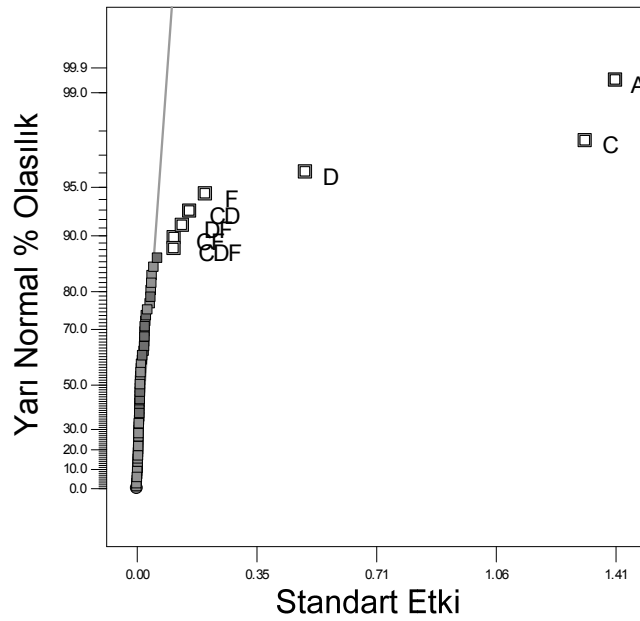
Şekil 4.2: Örnek bir Burkulma Mod Şekli

Şekil 4-2’de hazırlanan modelin örnek bir sonlu elemanlar simülasyonu sonucunda elde edilen birinci burkulma mod şekli görülmektedir. Veri üretirken sadece ilk moda ait kritik burkulma yükü kullanılacak olup mod şekli bu çalışmada göz önünde bulundurulmayacaktır.

4.2 Deney Tasarımı ile Modelleme

Veri üretme aşaması daha önce bahsedildiği gibi deney tasarımında ve yapay sinir ağlarında farklı olarak gerçekleşmektedir. Deney tasarımı yaklaşımında limit değerlerin kombinasyonlarına ihtiyaç duyulmaktadır ve tam faktöriyel tasarımında bütün kombinasyonlar kullanılmaktadır.

Bu bölümde çeyrek 2k analizi gösterilecek olup, merkezi kompozit yaklaşımının sadece sonuçlarından bahsedilecektir.



Şekil 4.3: Yarı-Normal Grafik

Çeyrek 2k tasarımı için, belirlenen 8 parametrenin en düşük ve en yüksek seviye kombinasyonlarından 64 tanesi gereklidir. Bu değerlerle simülasyon yapıldıktan sonra ilk olarak etkin rol oynayan parametrelerin ve parametre etkileşim terimlerinin tespiti gereklidir. Bunun için kullanılan yarı normal grafiği⁴ Şekil 4-3’de

⁴ 2. Bölümde normal etki grafiğinden bahsedilmişti. Yarı normal grafiğin de kullanımı benzerlikle birlikte etkin terimler farklı olarak grafiğin sağ tarafına yönelmektedir.

görülmektedir. Bu grafikte lineer eğilime sahip noktalar dışındaki noktalar modele etkisi yüksek olan terimlerdir. Model bu terimlerle oluşturulacaktır.

Grafikte A, C, D, F, CD, DF, CF ve CDF'nin etkin terimler olduğu görülmektedir. Bu harfler aslında parametrelerin gerçek değerlerinin $[-1,+1]$ aralığına normalize edilmiş halleridir.

Buradan hareketle yapılan regresyon sonucunda model,

$$\begin{aligned} \text{Log}_{10}(\text{kritik yük}) = & 3.95425 - 0.17676 \cdot \text{pw} + 269.22339 \cdot \text{pt} + 0.23827 \cdot \text{nos} \\ & + 45.53060 \cdot \text{swt} - 29.52378 \cdot \text{pt} \cdot \text{nos} - 10249.69301 \cdot \text{pt} \cdot \text{swt} - 11.86845 \cdot \text{nos} \cdot \text{swt} \\ & + 5267.61955 \cdot \text{pt} \cdot \text{nos} \cdot \text{swt} \end{aligned} \quad (4.1)$$

olarak elde edilir. Burada pw panel genişliğini, pt panel kalınlığını, nos stifner sayısını, swt de stifner gövdesi kalınlığını göstermektedir. Kritik yük, simülasyon çıkış değerleri arasındaki mertebe farklılıklarından dolayı artık dağılımını normalleştirmek adına logaritma fonksiyonu içinde alınarak regresyon gerçekleştirilmiştir.

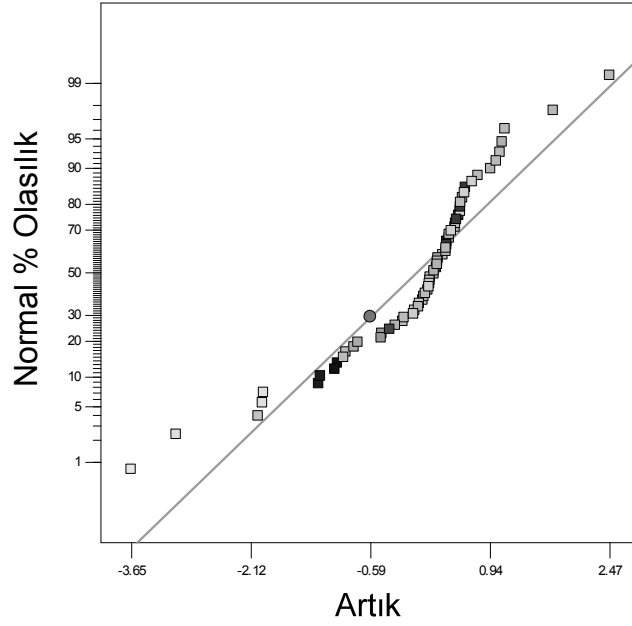
Elde edilen modelin geçerliliğini test etmek için yapılan ANOVA analizi sonuçları da Tablo 4-2'de görülmektedir.

Tablo 4.2: ANOVA Tablosu

Kaynak	Kareler Toplamı	df	Ortalama Karesi	F Değeri	p-değeri
Model	65.649	8	8.206	1171.526	< 0.0001
A-A	31.938	1	31.938	4559.589	< 0.0001
C-C	28.024	1	28.024	4000.723	< 0.0001
D-D	3.966	1	3.966	566.160	< 0.0001
F-F	0.656	1	0.656	93.641	< 0.0001
CD	0.390	1	0.390	55.671	< 0.0001
CF	0.195	1	0.195	27.858	< 0.0001
DF	0.288	1	0.288	41.095	< 0.0001
CDF	0.192	1	0.192	27.475	< 0.0001
Artık	0.385	55	0.007		
Toplam	66.034	63			

Tabloda p-değerlerinin çok düşük olduğu görülmektedir. Bu da hem modelin geçerli olduğu hem de seçili parametrelerin modele katkı sağladığı sonucuna varabilmemizi sağlar.

Oluşturulan modelin ne kadar uygun olduğu ile ilgili bir inceleme yapmak adına artık değerleri analizi yapmak gerekmektedir. Bununla ilgili olarak Şekil 4-4'te artık değerlerin normal olasılık grafiği görülmektedir.



Şekil 4.4: Artık Normal Olasılık Grafiği

Grafikte doğrusala yakın bir eğilim görülmektedir. Daha önce bahsedilen merkezi kompozit tasarımı uygulandığında bu eğilimin daha doğrusal olduğu gözlenmiştir. Fakat bir şeyi gözden kaçırmamak gerekir ki, o da merkezi kompozit tasarımında tam faktöriyel için gerekli 256 veri ve bunun yanında 2 derece etkileri de modele dâhil etmek için kullanılan 17 adet veriyle birlikte toplam 273 tane veri kullanılmıştır.

Sonuçta modeli oluşturmak için kullanılan verilerden bağımsız, rasgele olarak üretilen simülasyon sonuçlarıyla karşılaştırıldığında çeyrek faktöriyel tasarım 64 adet veriyle ortalama %22, 273 veriyle merkezi kompozit tasarımı da %17 ortalama hata üretmişlerdir.

Bir sonraki bölümde model yapay sinir ağlarıyla oluşturulacaktır ve her iki yöntemin sonuçları da beraber olarak takip eden bölümde gösterilecektir.

4.3 Yapay Sinir Ağlarıyla Modelleme

Şimdi de problemin yapay sinir ağlarıyla modellenmesi ele alalım. Öncelikle eğitim verilerinin elde edilmesi gerekmektedir. Fakat bu daha önceden belirlenmiş dizayn parametre kombinasyonları kullanılarak değil, parametrelerin tanım aralıklarında rasgele üretilmiş değerlerle elde edilecektir.

Üretilen rasgele değerler düzgün dağılıma sahiptir. Alternatif denemelerde, örneğin bir gauss dağılımı kullanılarak üretilen değerlerle yapılan eğitimin düzgün

dağılımdaki kadar efektif olmadığı görülmüştür. Bu da doğaldır, çünkü belli bir bölgeye kümelenmiş değerler kullanılırsa, diğer bölgelerde çok daha az bilgi alınır ve bu da genelleme özelliğine zarar verir.

Bu noktada, kullanılan veri sayısının bir parametre olarak alınıp, değişik sayıda veriler için yapay sinir ağlarının hata performansının bölümün sonunda gösterileceğini hatırlatmak isterim.

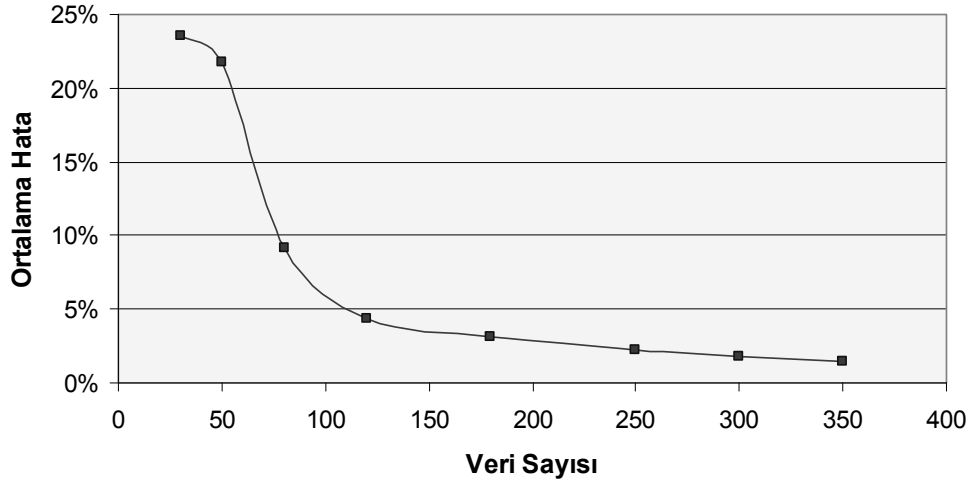
Kullanılan yapay sinir ağı, Mathworks Inc. (2006)'in geliştirdiği MATLAB adlı yazılımın “Neural Networks Toolbox” adlı alt paket programı ile oluşturulmuş olup, 1 adet gizli katman ve 1 adet 1 nöronlu çıkış katmanından oluşmaktadır. Gizli katmandaki nöronların aktivasyon fonksiyonu sigmoid, çıkış katmanında ise lineer bir fonksiyondur. Gizli katmandaki nöron sayısı da performansa doğrudan etki ettiği için tıpkı veri sayısı gibi parametrik olarak incelenmiştir.

Giriş-çıkış değerlerine önceki bölümde bahsedilen nedenlerden dolayı dönüşüm uygulanmış olup bu dönüşüm giriş değerleri için $[-0,9; +0,9]$ aralığına normalizasyon şeklinde olup, çıkış değerleri için ise logaritmik dönüşüm olarak gerçekleşmiştir.

Öğrenme algoritması olarak daha hızlı sonuçlar vermesinden ötürü Levenberg-Marquardt yöntemi tercih edilmiştir.

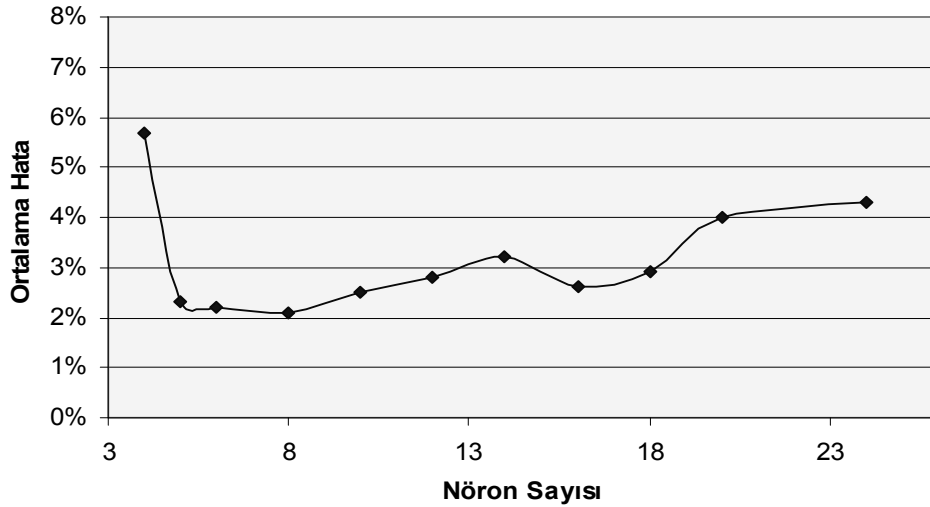
Ayrıca genelleştirme performansını arttırmak üzere ağ budama yaklaşımı kullanılmıştır. Bu yaklaşımın bir çeşidi olan ağırlık azaltma yönteminin MacKay (1991)'in geliştirdiği ve “Bayezyen Çıkarımı” esasına dayanan ve böylelikle performans fonksiyonundaki katsayıların otomatik olarak belirlendiği daha gelişmiş bir versiyonu, öğrenme algoritmasına yerleşik olarak değerlendirilmiştir.

Sonuçlara gelince, öncelikle öğrenme için kritik olan, kullanılan veri sayısının ortalama hataya etkisi Şekil 4-5'te gösterilmiştir. Buradaki sonuçlar gizli katmanında 12 adet nöron bulunan bir ağa aittir. Görüldüğü üzere ağ, 30 adet veriyle yaklaşık olarak %23 ortalama hata vermiştir. Bu hata oranı 350 adet veri ile %1.5'a kadar düşmektedir.



Şekil 4.5: Veri Sayısının Performansa Etkisi

Nöron sayısının etkisini incelediğimizde karşımıza Şekil 4-6'teki gibi bir durum çıkmaktadır. Burada veri sayısının 200'de sabit tutulmuştur. Sonuçlar bize en az ortalama hatanın 8 adet nöronda elde edildiğini göstermektedir.



Şekil 4.6: Nöron Sayısının Performansa Etkisi

Fakat dikkat edilmesi gereken önemli bir husus da hata değerlerinde o kadar da önemli bir fark olmadığıdır. Hata 24 nöronla 8 nöron arasında %4.3'ten %2.1'e düşmektedir. Bunun sebebi genelleştirme için uygulanan ağ budaması yaklaşımıdır.

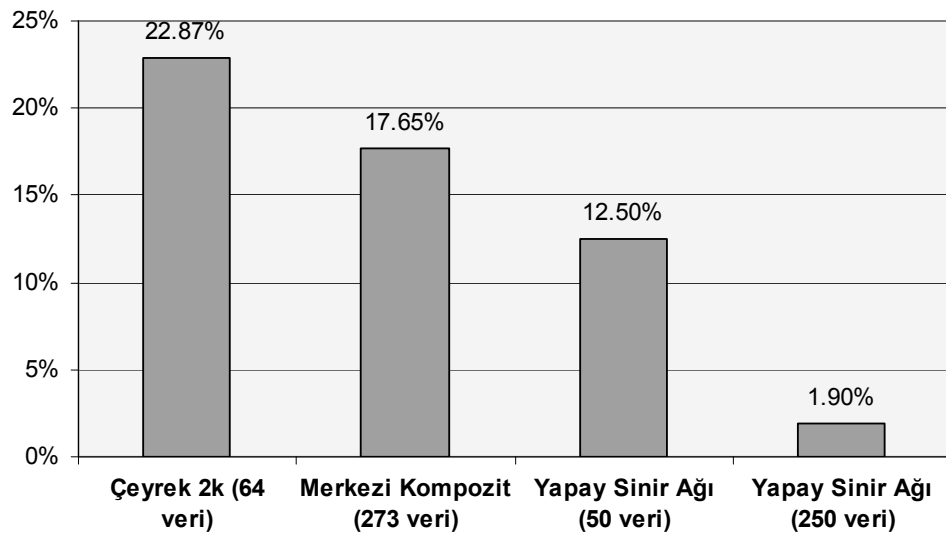
Gereksiz ağırlık değerleri bu yaklaşım sayesinde 0'a kadar yaklaştığından 24 nöronlu karmaşık sistemin ağırlık katsayılarından bir kısmı işlev görmez ve sistem aşırı uyum sorunundan daha az etkilenip daha iyi bir performans gösterir. Eğer ağ budama

kullanılmamış olsaydı nöron sayısının hata performansı açısından gözle görülür bir etkisi olurdu.

Ayrıca bu konuya benzer olarak veri sayısındaki artışın performansa olumlu etki etmesinin yanında veri sayısının mevcut mimarinin kaldırabileceğinden fazla olması durumunda ise yeterince değerlendirilemeyeceğini de söyleyebiliriz. Örneğin genelde 8 nöronlu ağ daha başarılı bir performans gösterse de 350 adet veri kullanıldığı zaman 12 nöronlu ağ daha iyi ortalama hata verebilmektedir.

4.4 Karşılaştırmalı Performans

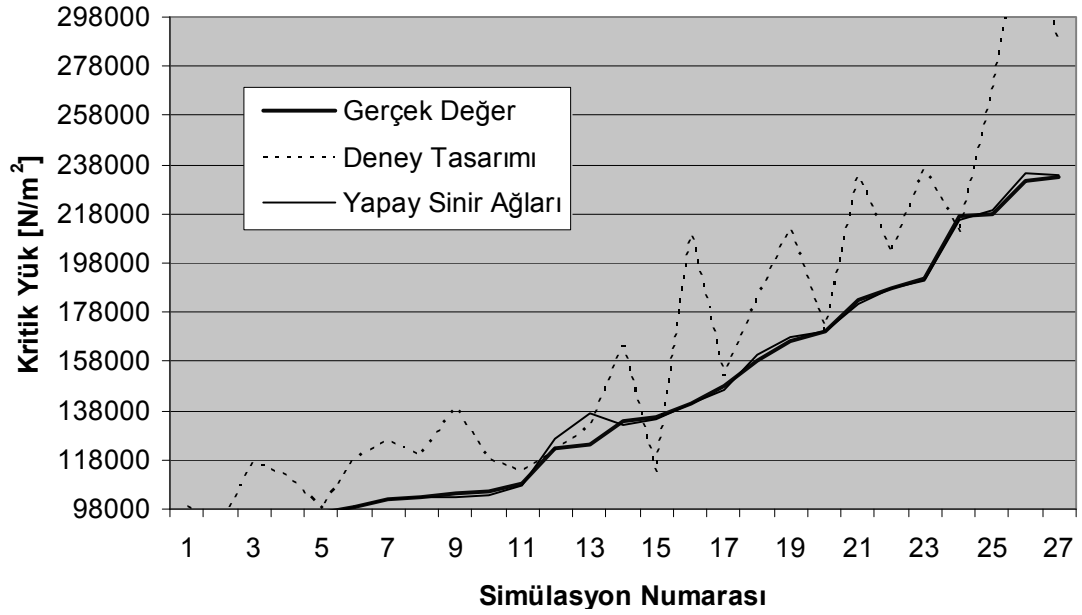
Önceki bölümlerde uygulamasından bahsedilen çeyrek 2k, merkezi kompozit tasarımı ve yapay sinir ağlarının ortalama hata performansları Şekil 4-7’de görülmektedir. Kullandığı veri sayısına göre de karşılaştırma yapılabilmesi için biri 50 diğeri 250 olmak üzere iki ayrı ağın performansı gösterilmiştir. Görüldüğü gibi 50 veriyle eğitilen ağ, 64 verinin kullanıldığı çeyrek 2k tasarımına, 250 veriyle eğitilen ağ da 273 veriyle oluşturulan merkezi kompozit tasarımına göre gözle görülür bir performans artışı göstermektedir.



Şekil 4.7: Hata Performans Değerleri

Hata tespitinin yapıldığı kontrol veri kümesinin bir kısmı için merkezi kompozit tasarımının ve yapay sinir ağının yaklaşık değeri ile simülasyon değeri birlikte Şekil

4-8’de gösterilmiştir. Burada da yapay sinir ağlarının simülasyon değerine çok yakın değerler verdiğini bir kez daha görmekteyiz.



Şekil 4.8: Örnek bir alt küme için gerçek ve tahmini değerler

Sonuç olarak hem yaklaşımın hassasiyetinin ayarlanabilirliği hem de bağlı performansı açısından yapay sinir ağları modelleme yöntemi olarak daha ön plana çıkmaktadır. Fakat şu da unutulmamalıdır ki buradaki asıl amaç daha önce de belirtildiği gibi bir genetik algoritma çalışmasında kullanılmak üzere model üretmedir ve bu modelin yüksek hassasiyette olma zorunluluğu yoktur. Yeter ki yaklaşım yapılacak duruma benzer formda bir model oluştursun yani ekstremum noktaları gerçeğe yakın olsun ki arama çalışması makul bir bölgeye doğru ilerlesin. Sonra daha hassas sonlu elemanlar simülasyonu seçilen dizaynın uygunluğunu kanıtlamak için kullanılabilir.

5. GELECEK ÇALIŞMALAR

Önceki bölümde yapay sinir ağlarının yüksek yakınlıkta tahminler yapabilen modelleme yaptığını gösterdik. Bir sonraki aşama olarak elde edilen modellerin bir genetik algoritma optimizasyonu çalışmasında kullanılıp asıl amacına yönelik performanslarının değerlendirilmesi gerekmektedir. Ayrıca yakınlık derecesini düşürülmesiyle de yani daha az veriyle de bu çalışmanın yürütölüp yürütölemeyeceğı kontrol edilmelidir.

Öte yandan aktif veri seçimi denen ve eğitim aşamasındaki ağa gönderilen her verinin etkisi değerlendirilerek ağa en fazla katkıyı yapacak giriş değerlerinin tespitine dayanan yaklaşımla daha az veriyle daha hassas modellerin oluşturulabilirliğı denenmelidir.

Örnek uygulama gerçek dizayn problemlerinin en fazla bir alt problemi olabilecek karmaşıklıktaydı. Daha büyük problemler için yapay sinir ağları modüler olarak üretilip, sonra da daha büyük bir ağ oluşturmak üzere birleştirilmek suretiyle modelleme yapılabilirliğı denenmelidir.

Böyle büyük bir problemi modelleyecek ağın yakınsaması arama uzayının büyüklüğünden zorlaşabilir. Bu çalışmada bahsedilen yöntemlere alternatif olarak böyle bir durumda ağırlık değerlerinin ve hatta ağ mimarisinin optimizasyonu için genetik algoritmalar düşünölmelidir.

KAYNAKLAR

- Abouhamze, M. ve Shakeri, M.,** 2006. Multi-objective stacking sequence optimization of laminated cylindrical panels using a genetic algorithm and neural networks, *Composite Structures*, Article in Press
- Ansys Inc.,** 2005. Release 10 Documentation for Ansys, Ansys Inc.
- Arora, J.S.,** 1989. Introduction to Optimum Design, McGraw-Hill, Singapore
- Box, G.E.P. and Behnken, D.W.,** 1960. Some New Three-Level Designs for the Study of Quantitative Variables, *Technometrics*, **2**, 455-475
- Box, G.E.P., Hunter, J.S. and Hunter, W.G.,** 2005. Statistics for Experimenters: Design, Innovation and Discovery, John Wiley & Sons, New Jersey
- Eiben, A.E. and Smith, J.E.,** 2003. Introduction to Evolutionary Computing, Springer-Verlag, New York
- Gambhir, M.L.,** 2004. Stability Analysis and Design of Structures, Springer-Verlag, Berlin
- Ganguli, R.,** 2002. Optimum design of a helicopter rotor for low vibration using aeroelastic analysis and response surface methods, *Journal of Sound and Vibration*, **258**, 327-344
- Gupta, M.M., Jin, L. and Homma N.,** 2003. Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory, John Wiley & Sons, New Jersey
- Hagan, M.T. and Menhaj, M.,** 1994. Training feed-forward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks* **5**, **6**, 989-993
- Haupt, R.L. and Haupt, S.E.,** 2004. Practical Genetic Algorithms, John Wiley, New Jersey
- Haykin, S.,** 1999. Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey
- Hornik, K., Stinchcombe, M. and White, H.,** 1989. Multilayer Feed-forward Networks Are Universal Approximators, *Neural Networks*, **5**, 359-366
- Kolmogorov, A.N.,** 1957. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition, *Dokl. Akad. Nauk USSR*, **114**, 953-956
- Lanzi, L. and Giavotto, V.,** 2006. Post-buckling optimization of composite stiffened panels: Computations and experiments, *Composite Structures*, **73**, 208-220

- LeCun, Y.**, 1993. Efficient Learning and Second-order Methods: A Tutorial, *Neural Information Processing Systems Conference*, Denver
- Levenberg, K.**, 1944. A Method for the Solution of Certain Problems in Least Squares, *Quart. Appl. Math.*, **2**, 164-168
- Lin, C.T. and Lee, C.S.G.**, 1996. Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice Hall, New Jersey
- Mackay, D.J.C.**, 1991. Bayesian Model Comparison and BackProp Nets, *Neural Information Processing Systems Conference*, Denver
- Marquardt, D.**, 1963. An Algorithm for Least-Squares Estimation of Nonlinear Parameters, *SIAM J. Appl. Math.*, **11**, 431-441
- Mathworks Inc.**, 2006. Matlab Release 2006b Documentation, Mathworks Inc.
- Mitchell, M.**, 1998. An Introduction to Genetic Algorithms, MIT Press, Massachusetts
- Myers, R.H. and Montgomery, D.C.**, 2002. Response Surface Methodology: Process and Product Optimization Using Designed Experiments, John Wiley & Sons, New York
- Todoroki, A. ve Ishikawa, T.**, 2003. Design of experiments for stacking sequence optimizations with genetic algorithm using response surface approximation, *Composite Structures*, **64**, 349-357
- Waszczyszyn, Z. ve Bartczak, M.**, 2002. Neural prediction of previous buckling loads of cylindrical shells with geometrical imperfections, *International Journal of Non-Linear Mechanics*, **37**, 763-775

ÖZGEÇMİŞ

Serdar Aytekin Köroğlu 1982 yılında İstanbul’da doğdu. İlköğrenimini Etiler Hasan Ali Yücel İlkokulu’nda tamamladıktan sonra ortaokul ve lise tahsilini Nişantaşı Anadolu Lisesi’nde yapmıştır. 2000 yılında İstanbul Teknik Üniversitesi Gemi İnşaatı Mühendisliği bölümünü kazanmış ve lisans derecesini 2004 yılında elde etmiştir. Bir sene süresince çeşitli tipte gemilerin yapı tasarımı üzerine çalıştıktan sonra 2005 yılı sonunda yüksek lisans eğitimine başlamıştır.