YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK KATMANLI ALGILAYICININ OPTİMİZASYONU VE AĞIRLIKLARA BAĞLI DUYARLIĞI

YÜKSEK LİSANS TEZİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSİ

LALE BAŞTÜRK

İSTANBUL 1993

ANNEME VE BABAMA ,

İÇİNDEKİLER

ÖNSÖZ

Lisansüstü çalışmalarım boyunca Yapay Nöron Ağları konusundaki eğitim ve öğretimi ile, bu tezin hazırlanmasında bana büyük destek olan, yol gösteren, cesaretlendiren Sayın Hocam Doç. Dr. Fikret GÜRGEN'e en derin saygı ve teşekkürlerimi sunarım.

ÖZET

Bu çalışmada geriye yayılım algoritması kullanılarak çok katmanlı algılayıcıda optimizasyon üzerine çeşitli problemler ele alınmış ve bu problemler üzerinde çok katmanlı algılayıcının ağırlıklara bağlı olan duyarlığı bir duyarlık ölçüsü için hesaplanmıştır.

Birinci bölümde yapay nöron ağ yapıları, işlem elemenları, yapay nöron ağlarında öğrenme konuları işlenmiştir. Bilgisayar programında kullanılan geriye yayılım öğrenme algoritması ve genelleştirilmiş delta kuralı açıklanmıştır.

İkinci bölümde xor, parite ve gauss deteksiyon problemleri üzerinde ileri yönlü çok katmanlı algılayıcının hata yakınsamasında çeşitli parametrelerin etkileri incelenmiştir ve yorumlanmıştır.

Üçüncü bölümde ise, ele alınan problemlerde her bir giriş paterni için diferansiyeli alınabilir aktivasyon fonksiyonlu tek çıkışlı çok katmanlı algılayıcının toplamsal ağırlık kusurları için duyarlığı üzerinde çalışılmıştır.

Sonuç bölümünde ise yapılan çalışmanın neticeleri tartışılmıştır.

SUMMARY

This thesis presents an empirical analysis of the effect of varius parametres on the error convergence of multilayer feedforward neural networks using the standart backpropagation learning algorithm and a sensitivity of multilayer neural networks for a weight set is calculated using a sensitivity measure.

In section 1; artificial neural network, processing elements and learning process are presented. Backpropagation algorithm which is used for computer programme; and generalized delta rule are explained.

In section 2; effect of various parametres on the error convergence of multilayer feedforward neural networks are examined on xor, parity and Gauss detection problems.

In section 3; a sensitivity caused by additive weight perturbations of a single-output multilayer perceptron with a differentiable activation function for each input patern is presented.

In conclusion section; results of presented thesis are explained.

BÖLÜM 1

YAPAY NÖRON AĞ YAPILARI

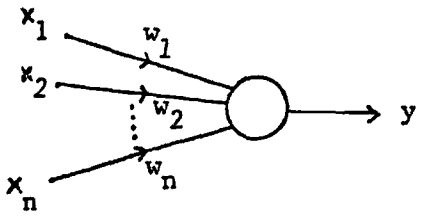
1.1 GİRİŞ:

İnsan beyni, gürültülü bir ortamda ve bozulmuş bilgi ile, görme, konuşma,bilgi düzeltmesi, şekil tanıma gibi işlevleri yerine getirebilir. Beynin işlem elemanları (nöronlar), milisaniyelerde, çağdaş elektronik ürünler ise nanosaniyelerde cevap verme yeteneğine sahiptirler. Bu gelişmeye rağmen, yapay zeka yöntemleri kullanılarak, sözkonusu yeteneklere, bilgisayarlar üzerinde ulaşılamaması, bilim adamlarını insan beynini incelemeye itmiştir.

Bir yapay nöron ağı,insan zekasının bazı fonksiyonlarını modellemek amacıyla, paralel olarak biraraya getirilmiş basit hesap birimleri (nöronlar) dizisidir. Konuşma ve görüntü tanıma alanlarında, insan gibi davranış gösterebilmek için geniş bir işlem yeteneğine ihtiyaç duyulması nedeni ile, yapay nöron ağları konusuna olan ilgi son yıllarda hızla artmıştır.Yapılan çalışmalarla varılan sonuçlara göre; yapay nöron ağlarında kullanılan nöron modeli, nöronları bir ağ yapısında birarada bağlama şekli ve ağırlıkların ayarlanması için kullanılacak öğrenme kuralının belirlenmesi önemlidir. Yapay nöron ağları ile hesaplama konusunda yapılan araştırmalar üç sınıfa girer. Birinci sınıfta, nöron ağının öğrenme özelliklerinin matemetiksel tanımı ve analizi yapılır. İkinci sınıfta, bu matemetiksel tanımları genişletmek, modellemek için bilgisayar simülasyonları kullanılır. Üçüncü sınıf ise, özel nöron fonksiyonlarını ya da nöron ağ sınıflarını gerçeklemeyi amaçlar.

1.2 YAPAY NÖRON YAPILARI:

Yapay nöron ağları, basit işlem elemanlarının paralel bir dizisidir.Her işlem elemanının yaptığı temel iş, giriş işaretlerinin bir dönüşümünü almaktır. Bir biyolojik nöronu temsil etmek için Şekil 2.1'deki işlem elemanı kullanılır.

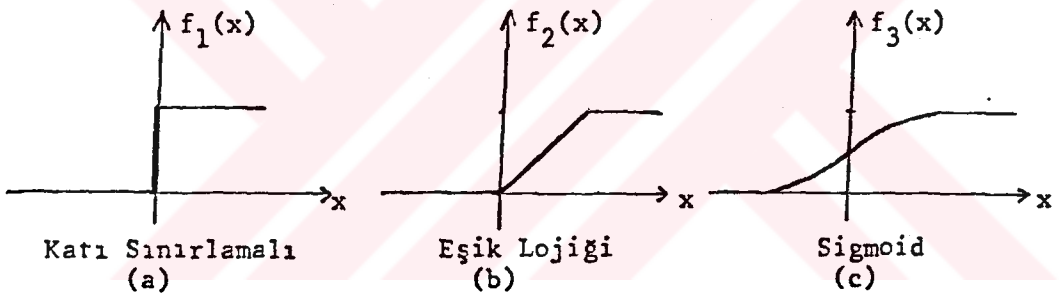


Şekil 1.1 Bir yapay nöron

Nöron modelinde y çıkışı;

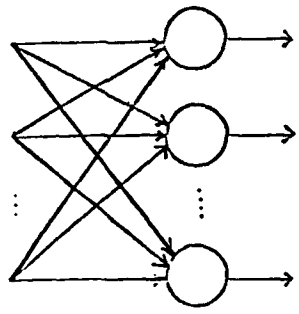
$$y = f(\sum_{i=1}^{n} w_i x_i - \theta) \qquad [1.1]$$

ile ifade edilir. Burada; (i=1,2,..,n) giriş işaretlerini, w_i x_i girişine karşı düşen ağırlığı gösterir. θ ise eşik değer olarak adlandırılır. f bir lineer olmayan fonksiyondur. Şekil 1.2'de f fonksiyonuna örnek üç tip lineer olmayan fonksiyon gösterilmiştir.



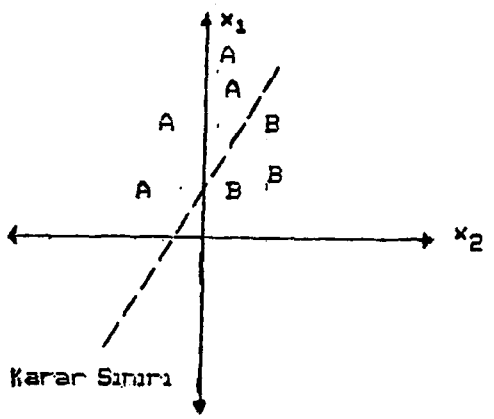
Şekil 1.2 Yapay nöron ağı modellerinde kullanılan üç tip lineer olmayan fonksiyon

(1.1) eşitliğinden görüldüğü gibi bir nöron modeli, n tane ağırlaştırılmış girişi toplamakta, bir eşik değerini bu toplamdan çıkartıp, sonucu bir lineer olmayan fonksiyondan geçirmektedir. Nöron ağlarının temel yapısı Şekil 1.3'deki tek katmanlı yapıdır. Tek katmanlı algılayıcılar sürekli değerli ve ikili girişin her ikisiyle de kullanılabilir.



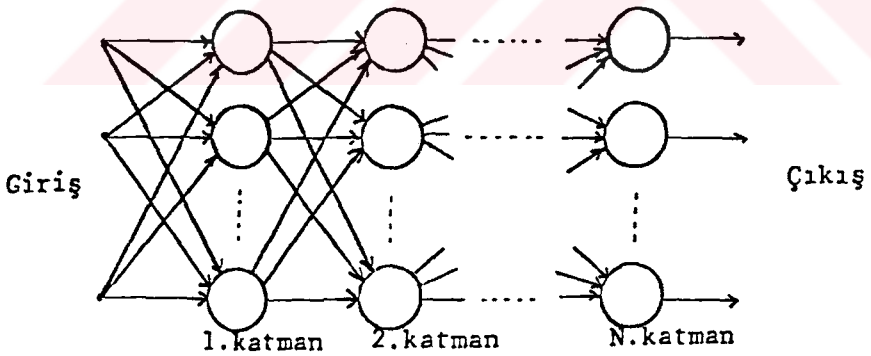
Şekil 1.3 Tek katmanlı ağ

Her algılayıcı bir girişin Şekil 1.4'de gösterilen A ve B ile işaret edilmiş iki sınıftan biriyle ilişkili olduğuna karar verir. Çıkış ya +1 ya da -1 dir. Eğer çıkış +1 ise girişin A sınıfına ait olduğuna, eğer çıkış -1 ise girişin B sınıfına ait olduğuna karar verilir [2].



Şekil 1.4

1960'larda Widrow, Hoff ve diğerleri tarafından oluşturulan tek katmanlı ağların çeşitli uygulamalarda yeteneklerinin sınırlı kaldığı ispatlanmıştır.Bu nedenle iki ya da daha fazla nöron katmanı kaskat bağlanarak Şekil 2.5'de gösterilen çok katmanlı ağlar oluşturulmuştur [6].

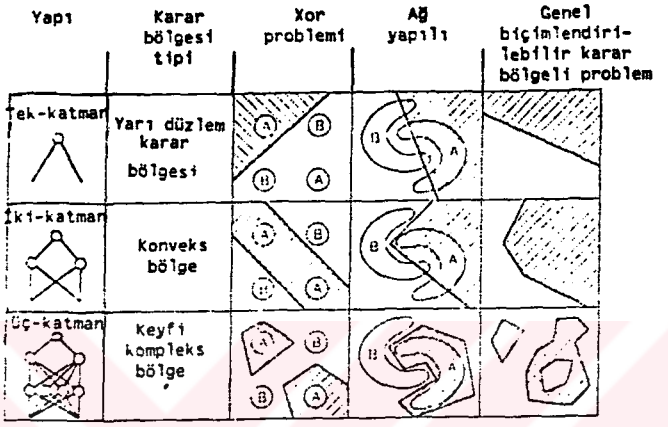


Şekil 1.5 Çok katmanlı ağ

Çok katmanlı ağların çalışması tek katmanlı ağların çalışmasına benzer. Her katmanın çıkışı, önceki katmanın çıkışlarının ağırlıklı toplamından üretilir.Çok katmanlı algılayıcılarda giriş ve çıkış katmanı arasında yer alan katmanlar gizli katman olarak adlandırılır. Çok katmanlı algılayıcılar tek katmanlı algılayıcıların birçoğunun üstesinden gelebilirler. Bir,iki ve üç katmanlı algılayıcıların yetenekleri Şekil 2.6'da tanımlanmıştır [2].

Bu şekildeki ikinci kolon, farklı ağlarla biçimlendirilmiş karar

bölgelerinin tiplerini gösterir. Sonraki iki kolon, xor problemi ve bir ağ yapılı bölgeli problem için biçimlendirilmiş karar bölgeleri örneklerini gösterir.En sağdaki kolon, en genel biçimlendirilebilir karar bölgeleri örneklerini verir. Bir katman algılayıcı yarıdüzlem karar bölgelerini biçimlendirir. Bir iki katmn algılayıcı uzaydaki konveks bölgeleri biçimlendirebilir. Üç katmanlı algılayıcı ise, keyfi komplex karar bölgelerini düzenleyebilir.



Şekil 1.6

1.3 İŞLEM ELEMANLARI

Bir yapay nöron ağı birçok işleme elemanının birarada bağlanmasından oluşmuştur. Her işlem elemanı istenildiği sayıda giriş bağlantısı alabilir. Fakat her işlem elemanında tek bir çıkış bağlantısı olabilir. Bu bağlantı kopya edilebilir.

İşlem elmanının bir yerel belleği vardır. Her elemanın yaptığı işlem transfer fonksiyonu ile belirlenir. Transfer fonksiyonları sürekli ya da kısmen sürekli olabilir. Kısmen sürekli çalışma konumunda 'aktif' halde eleman bir çıkış işareti verir.

Ayrıca nöron ağı bir takım alt kümelere ayrılır. Bu alt kümelerdeki elemanların transfer fonksiyonları aynıdır. Bu küçük gruplara 'katman' adı verilir. İşlem elemanları kendi aralarında haberleşebilir ve çıkış istenilen bir noktadan alınabilir[9].

1.4 ÖĞRENME

Öğrenme kuralı, giriş işareti ve transfer fonksiyonu ile bulunan değerlere göre ağırlıkların hepsini veya bir kısmını değiştiren bir denklemdir.

Yapay nöron ağları programlama yerine, örnekler ile eğitilir. Eğitme denetimli ve denetimsiz olmak üzere iki şekilde olur. Her ikisinde de , ağ seri denemelerden sonra çalışır. Denetimli öğrenmede, ağa hem giriş, hem de istenen çıkış bilgisi girilir. Her denemeden sonra ağ kendi çıkışını doğru cevaplar ile karşılaştırır ve çıkış hatası yeterince azalıncaya kadar, ağırlıklarını değiştirerek iterasyon yapar. Denetimsiz öğrenmede istenen çıkış yoktur. Giriş vektörü sisteme uygulanır ve sistem eğitimden önce tahmin edilemeyen uygun bir çıkış üretmek için kendisini öz-örgütler.

Her iki eğitmeden sonra da, ağ gerçek girişleri işlemek için hazırdır.

1.5 GERİYE YAYILIM ÖĞRENME ALGORİTMASI

Geriye yayılım öğrenme algoritması, Şekil 2.5 de gösterilen çok katmanlı algılayıcıların eğitilmesinde sıkça kullanılan bir algoritmadır.

Herhangi bir katmandaki j.ci birime gelen net giriş, bir önceki katmandaki birimlerin y_i çıkışlarının ağırlıklı toplamıdır.

$$net_j = \sum_{i=1}^{n} w_{ji}\, y_i \qquad [1.2]$$

İşlem elemanı, bu toplam girişi, türevi alınabilir, sürekli ve lineer olmayan bir fonksiyondan geçirir. Bu nedenle en çok kullanılan fonksiyon, sigmoid fonksiyonudur.

$$y = \frac{1}{1+e^{-(net_j)}} \qquad [1.3]$$

Bu fonksiyon y çıkışını Şekil 2.2c'de görüldüğü gibi 0 ve 1 değerleri arasında sınırlar. Geriye yayılım öğrenme algoritmasında girişe uygulanan bir giriş vektörüne karşı bir çıkış vektörü elde edilir. Bu çıkış vektörü, sisteme daha önceden verilen bir 'hedef' vektörü ile karşılaştırılır. Bu iki vektör arasındaki yeterince küçük olacak şekilde ağırlıklar ayarlanır.

Eğitme iki adımda olur.Birinci adımda, ağa bir giriş vektörü uygulanır ve her işlem elemanının çıkışı hesaplanır. Elde edilen çıkışlar bir sonraki denemede ağırlıkların ayarlanmasında kullanılmak üzere saklanır. Ikinci adımda, çıkış ve hedef vektörü arasındaki farka göre hesaplanan çıkış hatası geriye doğru iletilir. Ağ ağırlıkları bu hataya göre ayarlanır. Bu işlem, yeterince doğru cevaplar alınıncaya kadar tekrar edilir [6].

1.6 GENELLEŞTİRİLMİŞ DELTA KURALI

Nöron ağı öğrenme işlemini ağırlıkları değiştirerek yapar. 'p' giriş paterni için ağırlıklardaki değişim

$$\Delta_p w_{ji} = \eta (t_{pj} - O_{pj}) i - \eta \delta_{pj} i_{pi} \qquad [1.4]$$

ile verilir.

Burada, t_{pj} p paterni için hedef çıkışın j. bileşenini, O_{pj} p paterni için aktif çıkışın j.bileşenini, i_{pi} p giriş paterninin i. elemanının değerini gösterir. Geriye yayılım algoritmasında kullanılan hata kriteri karesel ortalama hatadır. Buna göre p paterni için oluşan hata,

$$E_p = \frac{1}{2} \sum_j (t_{pj} - O_{pj})^2 \qquad [1.5]$$

ve tüm paternler için toplam hata,

$$E = \sum_p E_p \qquad [1.6]$$

dir.

Sistemde ağırlık değişimi, oluşan hataya göre yapıldığı için ' E_p / w_{ji} ' ifadesini incelemeliyiz. Bu inceleme ağın gizli katmana sahip olup olmamasına bağlı olarak değişik sonuç verir.

A- Nöron ağında gizli katman yoksa, zincir kuralına göre

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial O_{pj}} \frac{\partial O_{pj}}{\partial w_{ji}} \qquad [1.7]$$

dir.

$$O_{pj} = \sum_i w_{ji} i_{pi} \qquad [1.8]$$

olduğundan

$$\frac{\partial O_{pj}}{\partial w_{ji}} = i_{pi} \qquad [1.9]$$

olur.

[1.5] eşitliğinden

$$\frac{\partial E_p}{\partial O_{pj}} = -\delta_{pj} \qquad [1.10]$$

dir.

Sonuç olarak,

$$\frac{\partial E_p}{\partial w_{ji}} = -\delta_{pj} i_{pi} \qquad [1.11]$$

B- Nöron ağında gizli katman varsa, zincir kuralına göre

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} \qquad [1.12]$$

dir.

$$net_{pj} = \sum_i w_{ji}\, O_{pi} \qquad [1.13]$$

olduğundan

$$\frac{\partial net_{pj}}{\partial w_{ji}} = O_{pi} \qquad [1.14]$$

olur.

$$O_{pj} = f_j(net_{pj}) \qquad [1.15]$$

ve

$$\delta_{pj} = \frac{-\partial E_p}{\partial net_{pj}} \qquad [1.16]$$

olmak üzere

a- O_{pj} yapay nöron ağı çıkışı ise

$$\delta_{pj} = (t_{pj} - O_{pj})\, f(net_{pj}) \qquad [1.17]$$

olur.

b- O_{pj} gizli katman çıkışı ise

$$\delta_{pj} = f(net_{pj}) \sum_k \delta_{pk}\, w_{pk} \qquad [1.18]$$

olur.

Geriye yayılım algoritmasında öğrenme, yani ağırlık değişimi ' E_p / w_{pj} ' ile orantılı olarak yapılır. Orantı sabiti ; öğrenme oranıdır. Genellikle 0 ile 1 arasında seçilir. Bu sabit ne kadar büyük ise ağırlıktaki değişime o kadar büyüktür. Pratikte osilasyona yol açmayacak büyüklükte öğrenme oranı seçilir. Öğrenme oranının daha

büyük değerlerine imkan sağlayarak hata yakınsamasını hızlandırmak için; genelleştirilmiş delta kuralı, momentum terimi ile modifiye edilir [1,8].

$$\Delta w_{ji}(t+1) = \eta\, \delta_{pj}\, O_p + \alpha\, \Delta w_{ji}(t) \qquad [1.19]$$

Geriye yayılım öğrenme algoritması Tablo 1.1'de gösterilmiştir.

Geriye yayılım algoritmasının bazı temel sorunları vardır. Bu sorunlar eğitimin yavaş olması, algoritmanın yerel minimuma yakalanabilmesi, otonom olmamasıdır.

Geriye yayılım algoritmasında eğitim yavaştır. Ardarda iki katmandaki n tane nöron için, n tane ağırlık eğitilmektedir. Hatanın geriye yayılımı da bu ağırlıklar üzerinden yapılır. Bu işlemler oldukça zaman alır.

Hatayı minimum yapacak ağ ağırlıkları, bir gradyent arama tekniği kullanılarak bulunur. Eğitme sırasında yerel minimuma yakalanma sorunu vardır. Bu durumda ağ kabul edilemeyecek bir çözüme yakınsar.Bu sorundan kaçınmak için öğrenme oranı azaltılabilir.

Ağa yeni bilgiler öğretildiğinde eskiden öğrenilen bilgilerin unutulması nedeniyle; ağ,değişen çevre şartlarına uyum sağlanmasının beklendiği durumlar için uygun değildir.

TABLO 1.1

GERİYE YAYILIM AĞININ EĞİTME ALGORİTMASI

ADIM 1 Tüm ağırlıkların ilk değerlerini belirle.
Tüm ağırlıkları küçük raslantısal değerlere ayarla.

ADIM 2 Girişi ve istenen çıkışı uygula.
Sürekli değerli bir $x = [x_1\ x_2\ \ldots\ x_n]^T$ giriş vektörünü uygula ve istenen $d = [d_1\ d_2\ \ldots\ d_n]^T$ çıkış vektörünü belirle. Her denemede yeni bir giriş uygulanır veya ağırlıklar kararlı oluncaya dek bir eğitme kümesinden örnekler tekrar uygulanabilir.

ADIM 3 Gerçek çıkışları hesapla.
En son katmandaki birimlerin çıkışlarını belirleyebilmek için her katmandaki birimlerin çıkışlarını sırayla belirle ve sakla.

ADIM 4 Ağırlıkları ayarla.
Çıkış katmanındaki birimlerden ilk katmandaki birimlere kadar, geriye doğru çalışan bir yinelemeli algoritma kullanılacaktır. Ağırlıklar [1.19] eşitliğine göre ayarlanır.

ADIM 5 Adım 2'ye giderek tekrar et.

BÖLÜM 2

ÇOK KATMANLI ALGILAYICININ PARAMETRELERİNİN OPTİMİZASYONU

Bu bölümde xor,parite ve gauss deteksiyon problemleri üzerinde, çok katmanlı algılayıcının hata yakınsamasında öğrenme oranı, momentum katsayısı ve gizli katmandaki düğüm sayısının etkileri incelenmiştir[5].

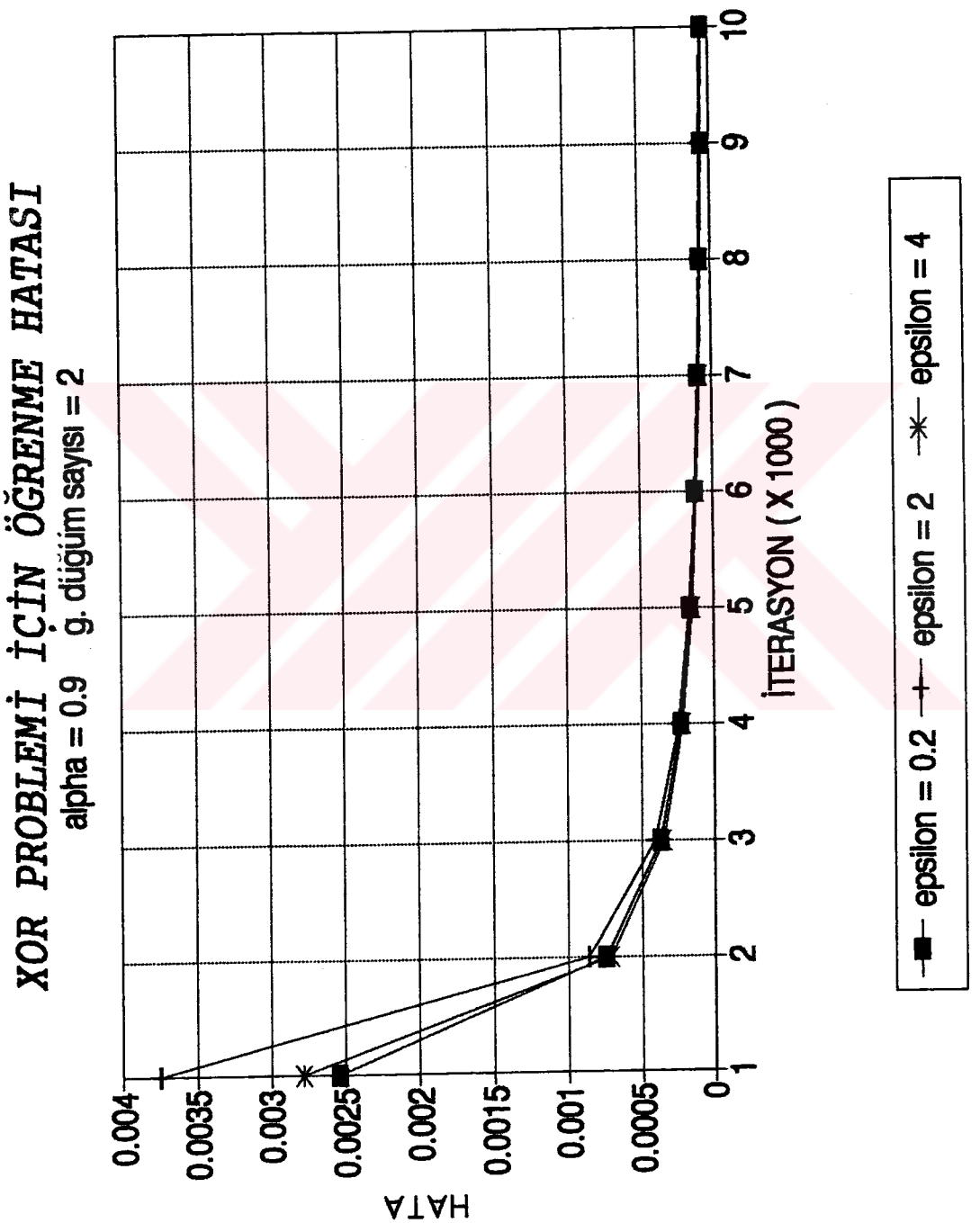
Xor ve parite problemlerinde kullanılan nöron ağı bir gizli katmana, gauss probleminde kullanılan ağ ise iki gizli katmana sahiptir.

Xor problemi için iki giriş düğümü, iki gizli düğüm, bir çıkış düğümü; üç bit parite problemi için üç giriş düğümü, üç gizli düğüm, bir çıkış düğümü; dört bit parite problemi için dört giriş düğümü, dört gizli düğüm ve bir çıkış düğümü kullanıldı.Parite problemlerinde giriş paternlerindeki birlerin sayısı tek ise sonuç 1, çift ise sonuç 0'dır.

Gauss deteksiyon probleminde kullanılan nöron ağında giriş ve çıkış düğüm sayısı bir, gizli düğüm sayısı ikidir. Bu problemde ağ, tek boyutlu -1 ve 2 beklendik değerli iki gauss dağılımı arasındaki farka göre eğitildi. Her iki gauss dağılımı için varyans 1'e eşittir. Deteksiyon için eşik değeri 0.5'dir. Giriş değerleri -1 ve 2 ortalama değerli gauss sayı üretecinden rasgele üretilmektedir. Üretilen sayı 0.5'den büyük ise 1, küçük ise 0 sonucu alınmaktadır [4].

2.1 ÖĞRENME ORANI

Geriye yayılım algoritmasında öğrenme oranı önemli bir parametredir.Öğrenme oranının farklı değerleri için hata değerleri hesaplanmıştır. Deneylerde momentum katsayısı 0.9 alınmıştır. Gizli katmandaki düğüm sayısı xor ve gauss deteksiyon problemi için iki, üç bit parite problemi için üç, dört bit parite problemi için dört alınmıştır. Elde edilen sonuçlara göre öğrenme oranının büyümesi hata yakınsamasını (küçülmesini) geciktirmektedir. Çünkü öğrenme oranı büyüdüğü takdirde ağırlık uzayında minimum nokta civarında osilasyon oluşur.


XOR PROBLEMİ İÇİN ÖĞRENME HATASI
alpha = 0.9 g. düğüm sayısı = 2
HATA
0.004
0.0035
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
epsilon = 0.2
epsilon = 2
epsilon = 4

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 2

	giriş		cıkış	hedef değer
0	0.100	0.100	0.105	0.100
1	0.100	0.900	0.896	0.900
2	0.900	0.100	0.896	0.900
3	0.900	0.900	0.105	0.100

iterasyon=10000 eps= 0.003386 hata_1= 0.000046
hata_t= 0.000046

epsilon = 2
alpha = 0.9
gizli düğüm sayısı = 2

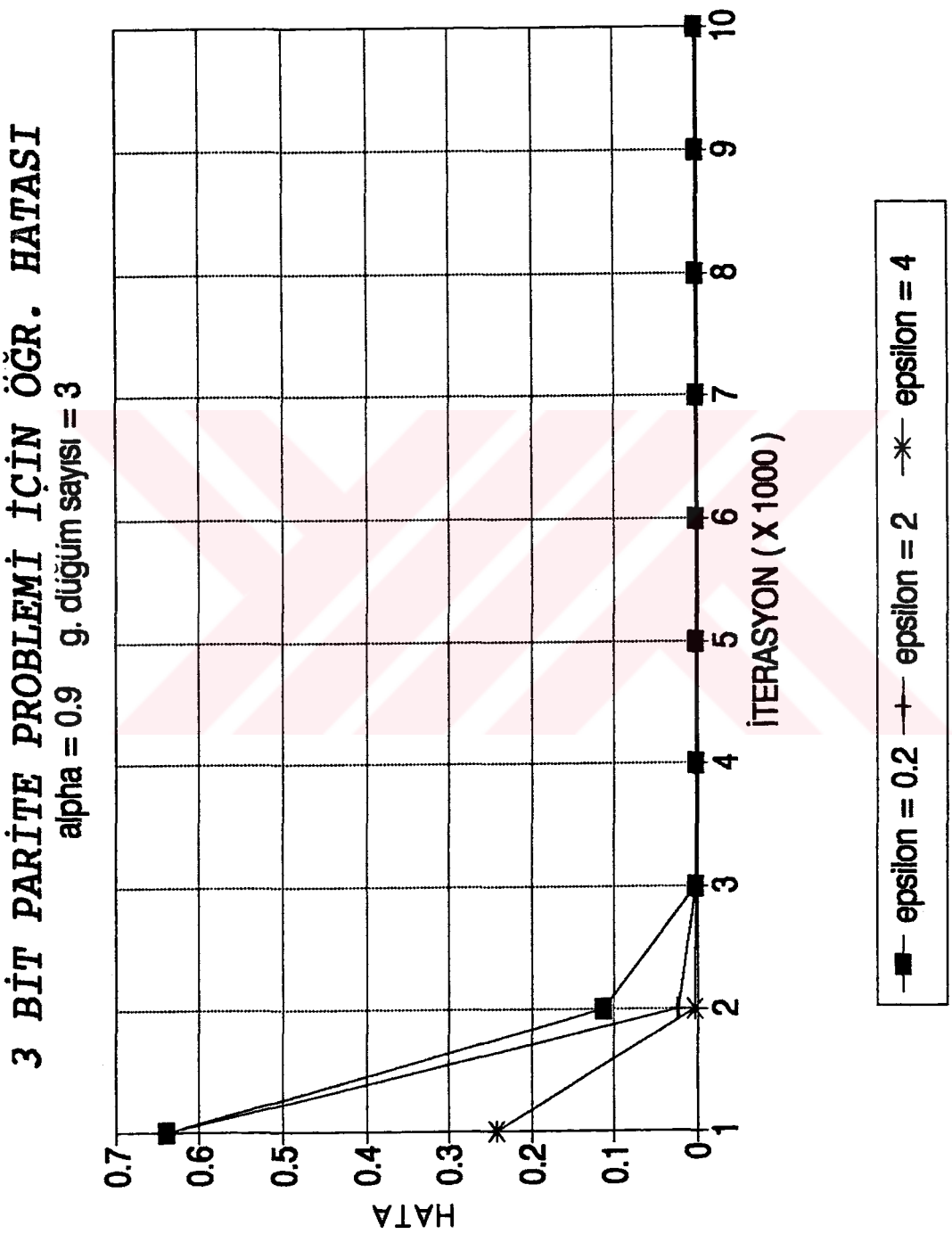
	giriş		cıkış	hedef değer
0	0.100	0.100	0.104	0.100
1	0.100	0.900	0.895	0.900
2	0.900	0.100	0.895	0.900
3	0.900	0.900	0.106	0.100

iterasyon=10000 eps= 0.003420 hata_1= 0.000047
hata_t= 0.000047

epsilon = 4
alpha = 0.9
gizli düğüm sayısı = 2

	giriş		cıkış	hedef değer
0	0.100	0.100	0.105	0.100
1	0.100	0.900	0.897	0.900
2	0.900	0.100	0.893	0.900
3	0.900	0.900	0.103	0.100

iterasyon=10000 eps= 0.003267 hata_1= 0.000043
hata_t= 0.000043


3 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
alpha = 0.9 g. düğüm sayısı = 3
HATA
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
epsilon = 0.2
epsilon = 2
epsilon = 4

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 3

	giriş			cıkış	hedef değer
0	0.100	0.100	0.100	0.104	0.100
1	0.100	0.100	0.900	0.896	0.900
2	0.100	0.900	0.100	0.900	0.900
3	0.100	0.900	0.900	0.099	0.100
4	0.900	0.100	0.100	0.896	0.900
5	0.900	0.100	0.900	0.109	0.100
6	0.900	0.900	0.100	0.099	0.100
7	0.900	0.900	0.900	0.900	0.900

iterasyon=10000 eps= 0.004133 hata_1= 0.000068
hata_t= 0.000068

epsilon = 2
alpha = 0.9
gizli düğüm sayısı = 3

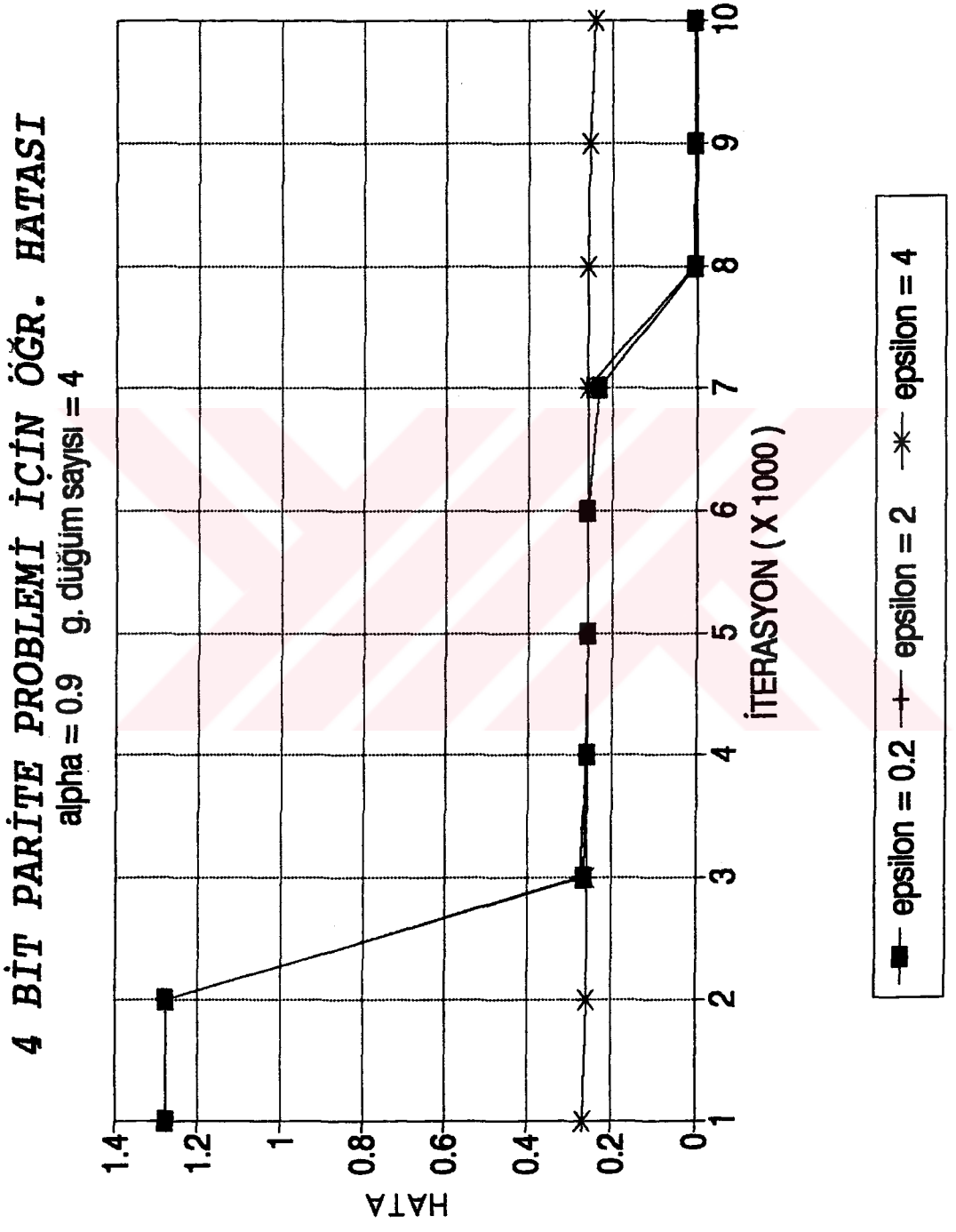
	giriş			cıkış	hedef değer
0	0.100	0.100	0.100	0.104	0.100
1	0.100	0.100	0.900	0.896	0.900
2	0.100	0.900	0.100	0.900	0.900
3	0.100	0.900	0.900	0.099	0.100
4	0.900	0.100	0.100	0.896	0.900
5	0.900	0.100	0.900	0.109	0.100
6	0.900	0.900	0.100	0.099	0.100
7	0.900	0.900	0.900	0.900	0.900

iterasyon=10000 eps= 0.004088 hata_1= 0.000067
hata_t= 0.000067

epsilon = 4
alpha = 0.9
gizli düğüm sayısı = 3

	giriş			cıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.100
1	0.100	0.100	0.900	0.899	0.900
2	0.100	0.900	0.100	0.899	0.900
3	0.100	0.900	0.900	0.103	0.100
4	0.900	0.100	0.100	0.899	0.900
5	0.900	0.100	0.900	0.103	0.100
6	0.900	0.900	0.100	0.103	0.100
7	0.900	0.900	0.900	0.890	0.900

iterasyon=10000 eps= 0.004120 hata_1= 0.000068
hata_t= 0.000068


4 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
alpha = 0.9 g. düğüm sayısı = 4
HATA
İTERASYON (X 1000)
epsilon = 0.2
epsilon = 2
epsilon = 4

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 4

	giriş				cıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.101	0.100
1	0.100	0.100	0.100	0.900	0.900	0.900
2	0.100	0.100	0.900	0.100	0.900	0.900
3	0.100	0.100	0.900	0.900	0.100	0.100
4	0.100	0.900	0.100	0.100	0.900	0.900
5	0.100	0.900	0.100	0.900	0.100	0.100
6	0.100	0.900	0.900	0.100	0.100	0.100
7	0.100	0.900	0.900	0.900	0.894	0.900
8	0.900	0.100	0.100	0.100	0.900	0.900
9	0.900	0.100	0.100	0.900	0.100	0.100
10	0.900	0.100	0.900	0.100	0.100	0.100
11	0.900	0.100	0.900	0.900	0.894	0.900
12	0.900	0.900	0.100	0.100	0.100	0.100
13	0.900	0.900	0.100	0.900	0.894	0.900
14	0.900	0.900	0.900	0.100	0.894	0.900
15	0.900	0.900	0.900	0.900	0.126	0.100

iterasyon=10000 eps= 0.010283 hata_1= 0.000423 hata_t= 0.000422

epsilon = 2
alpha = 0.9
gizli düğüm sayısı = 4

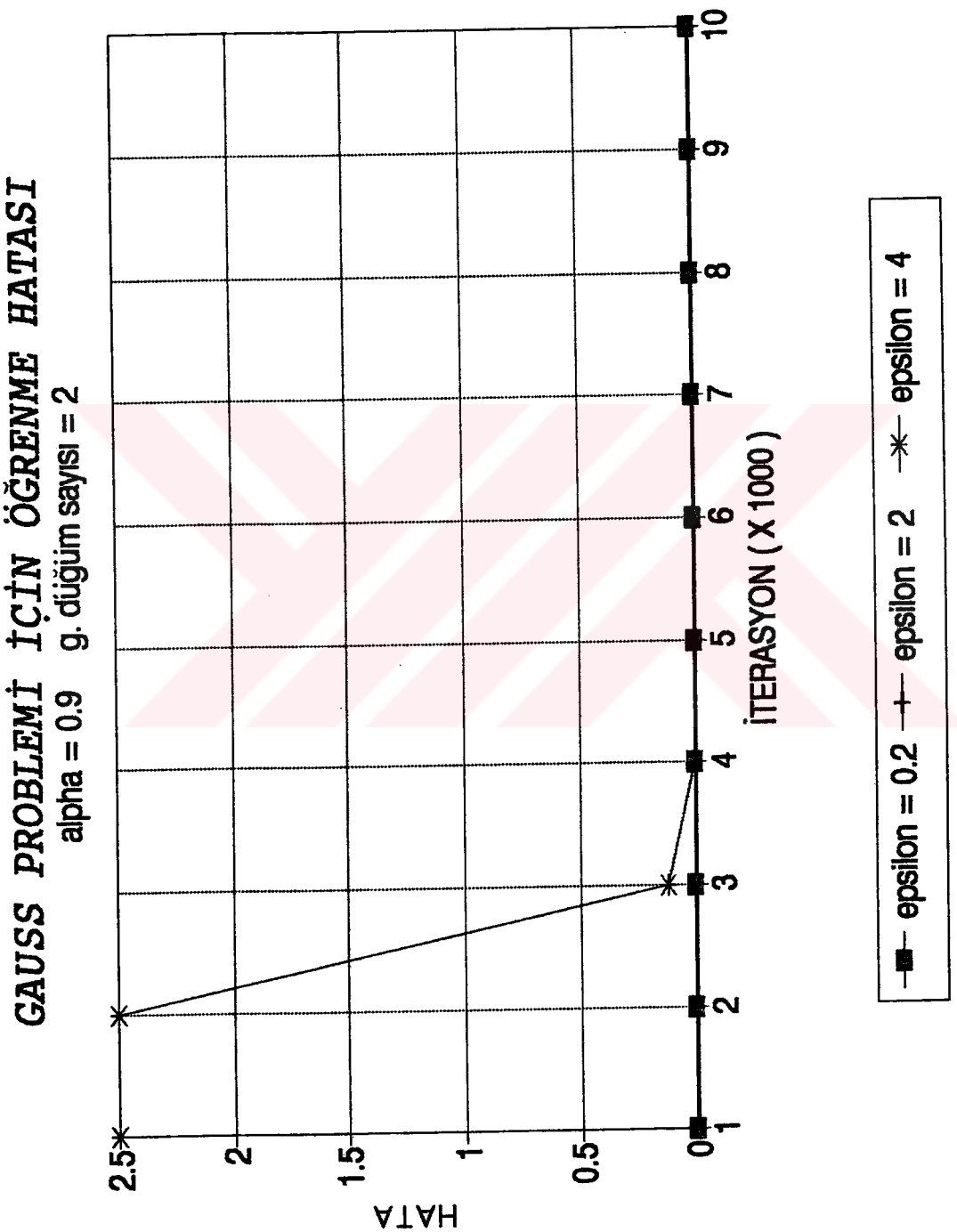
	giriş				cıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.101	0.100
1	0.100	0.100	0.100	0.900	0.900	0.900
2	0.100	0.100	0.900	0.100	0.900	0.900
3	0.100	0.100	0.900	0.900	0.100	0.100
4	0.100	0.900	0.100	0.100	0.900	0.900
5	0.100	0.900	0.100	0.900	0.100	0.100
6	0.100	0.900	0.900	0.100	0.100	0.100
7	0.100	0.900	0.900	0.900	0.893	0.900
8	0.900	0.100	0.100	0.100	0.900	0.900
9	0.900	0.100	0.100	0.900	0.100	0.100
10	0.900	0.100	0.900	0.100	0.100	0.100
11	0.900	0.100	0.900	0.900	0.893	0.900
12	0.900	0.900	0.100	0.100	0.100	0.100
13	0.900	0.900	0.100	0.900	0.893	0.900
14	0.900	0.900	0.900	0.100	0.893	0.900
15	0.900	0.900	0.900	0.900	0.128	0.100

iterasyon=10000 eps= 0.011184 hata_1= 0.000500 hata_t= 0.000499

epsilon = 4
alpha = 0.9
gizli düğüm sayısı = 4

	giriş				cıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.077	0.100
1	0.100	0.100	0.100	0.900	0.896	0.900
2	0.100	0.100	0.900	0.100	0.910	0.900
3	0.100	0.100	0.900	0.900	0.295	0.100
4	0.100	0.900	0.100	0.100	0.897	0.900
5	0.100	0.900	0.100	0.900	0.154	0.100
6	0.100	0.900	0.900	0.100	0.294	0.100
7	0.100	0.900	0.900	0.900	0.297	0.900
8	0.900	0.100	0.100	0.100	0.904	0.900
9	0.900	0.100	0.100	0.900	0.083	0.100
10	0.900	0.100	0.900	0.100	0.142	0.100
11	0.900	0.100	0.900	0.900	0.896	0.900
12	0.900	0.900	0.100	0.100	0.080	0.100
13	0.900	0.900	0.100	0.900	0.912	0.900
14	0.900	0.900	0.900	0.100	0.895	0.900
15	0.900	0.900	0.900	0.900	0.299	0.100

iterasyon=10000 eps= 0.249395 hata_1= 0.248792 hata_t= 0.242525


GAUSS PROBLEMİ İÇİN ÖĞRENME HATASI
alpha = 0.9 g. düğüm sayısı = 2
HATA
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
epsilon = 0.2
epsilon = 2
epsilon = 4

```
epsilon             = 0.2
alpha               = 0.9
gizli düğüm sayısı = 2

      giriş  cıkış     hedef değer
-----  -----  ------    -------------
    0  0.005  0.020        0.000
    1  -0.71  0.007        0.000
    2  -0.24  0.009        0.000
    3  -1.27  0.007        0.000
    4  -1.86  0.007        0.000
    5  1.009  0.995        1.000
    6  2.389  0.995        1.000
    7  0.532  0.983        1.000
    8  1.602  0.995        1.000
    9  3.574  0.995        1.000
iterasyon=10000     eps= 0.011353     hata_1= 0.000516
                                      hata_t= 0.000516
```

```
epsilon             = 2
alpha               = 0.9
gizli düğüm sayısı = 2

      giriş  cıkış     hedef değer
-----  -----  ------    -------------
    0  -1.25  0.008        0.000
    1  -0.70  0.010        0.000
    2  -1.61  0.007        0.000
    3  -1.09  0.008        0.000
    4  -0.95  0.008        0.000
    5  1.282  0.988        1.000
    6  2.603  0.989        1.000
    7  3.973  0.989        1.000
    8  2.968  0.989        1.000
    9  1.555  0.988        1.000
iterasyon=10000     eps= 0.011051     hata_1= 0.000489
                                      hata_t= 0.000489
```

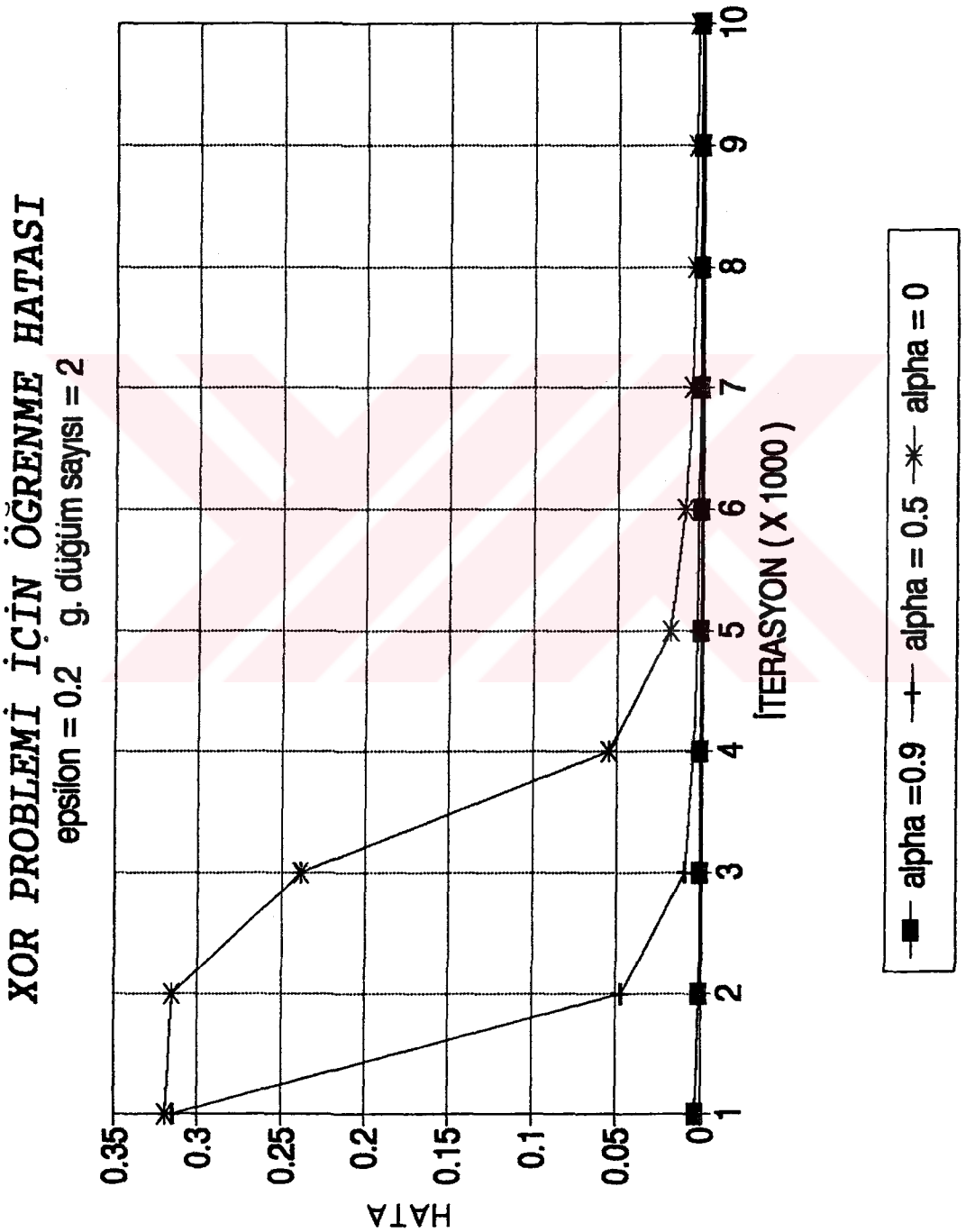
```
epsilon            = 4
alpha              = 0.9
gizli düğüm sayısı = 2

       giriş  cıkış     hedef deger
-----  -----  ------    -------------
    0  0.432  0.020        0.000
    1  -1.86  0.002        0.000
    2  -0.22  0.002        0.000
    3  -0.79  0.002        0.000
    4  -1.81  0.002        0.000
    5  2.121  0.993        1.000
    6  1.218  0.990        1.000
    7  2.543  0.993        1.000
    8  2.096  0.993        1.000
    9  0.983  0.977        1.000
iterasyon=10000     eps= 0.012296     hata_1= 0.000605
                                      hata_t= 0.000605
```

2.2 MOMENTUM KATSAYISI

Öğrenme oranının büyük değerleri için hata yakınsamasını hızlandırmak açısından momentum terimi oldukça önemlidir. Momentum terimi önceki iterasyonda elde edilen ağırlık değişiminin o andaki ağırlık değişimine etkisini gösterir.

Simülasyon sonuçlarına göre momentum katsayısı küçüldükce hata değeri büyümektedir. Bunun nedeni, momentum katsayısı küçük olduğunda bir önceki ağırlık değişiminin bir sonraki ağırlık değişimine etkisinin az olmasıdır. Analizde momentum katsayısı değiştirilirken, öğrenme oranı 0.2 alınmıştır.

Gizli katmandaki düğüm sayısı xor ve gauss problemleri için iki, üç bit parite problemi için üç, dört bit parite problemi için dörtdür.


XOR PROBLEMİ İÇİN ÖĞRENME HATASI
epsilon = 0.2 g. düğüm sayısı = 2
HATA
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
alpha =0.9
alpha = 0.5
alpha = 0

epsilon = 0.2
alpha = 0.5
gizli düğüm sayısı = 2

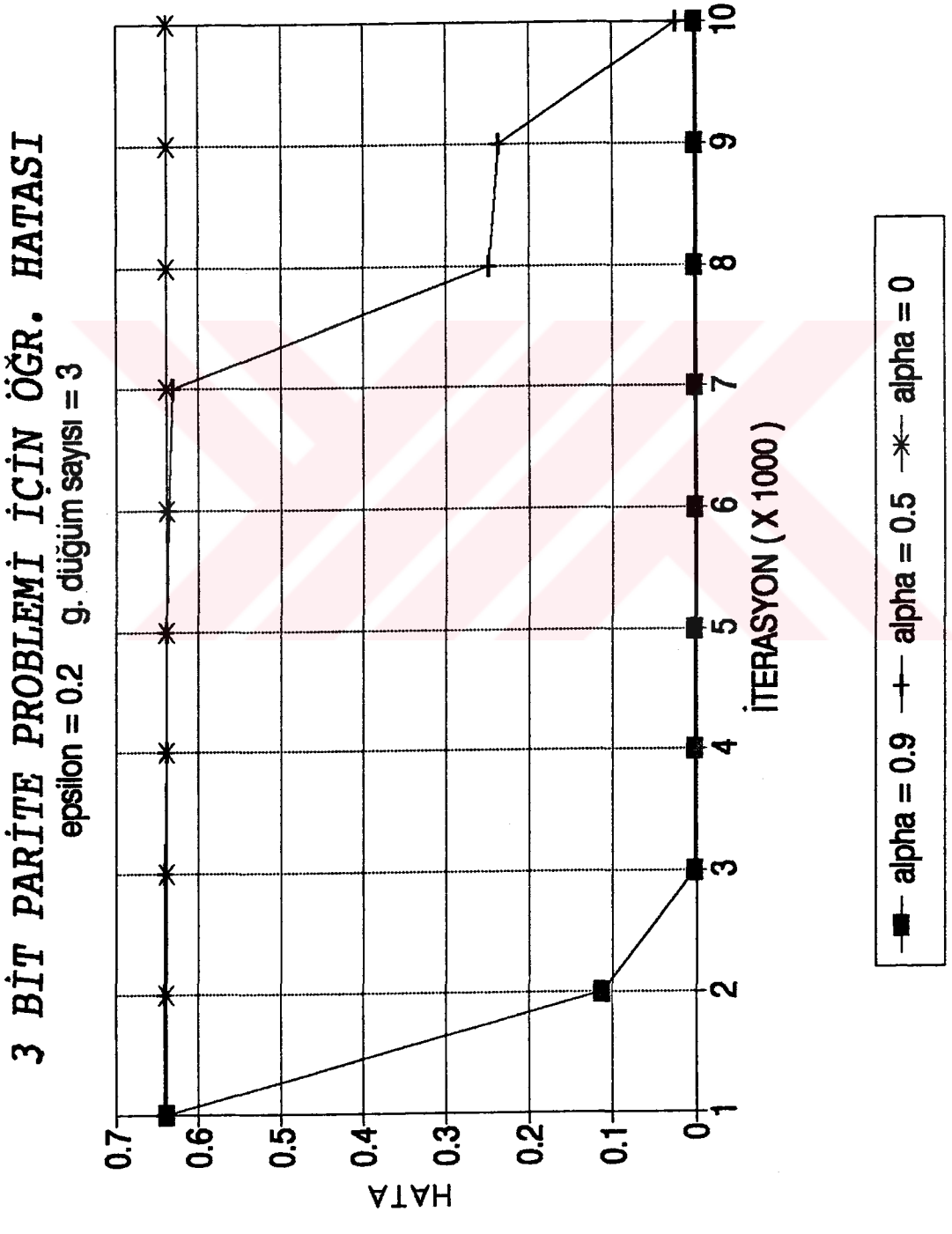
	giriş		cıkış	hedef değer
0	0.100	0.100	0.119	0.100
1	0.100	0.900	0.881	0.900
2	0.900	0.100	0.881	0.900
3	0.900	0.900	0.122	0.100

iterasyon=10000 eps= 0.013890 hata_1= 0.000772
hata_t= 0.000771

epsilon = 0.2
alpha = 0
gizli düğüm sayısı = 2

	giriş		cıkış	hedef değer
0	0.100	0.100	0.135	0.100
1	0.100	0.900	0.864	0.900
2	0.900	0.100	0.864	0.900
3	0.900	0.900	0.143	0.100

iterasyon=10000 eps= 0.026794 hata_1= 0.002872
hata_t= 0.002867


3 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
epsilon = 0.2 g. düğüm sayısı = 3
HATA
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
alpha = 0.9
alpha = 0.5
alpha = 0

epsilon = 0.2
alpha = 0.5
gizli düğüm sayısı = 3

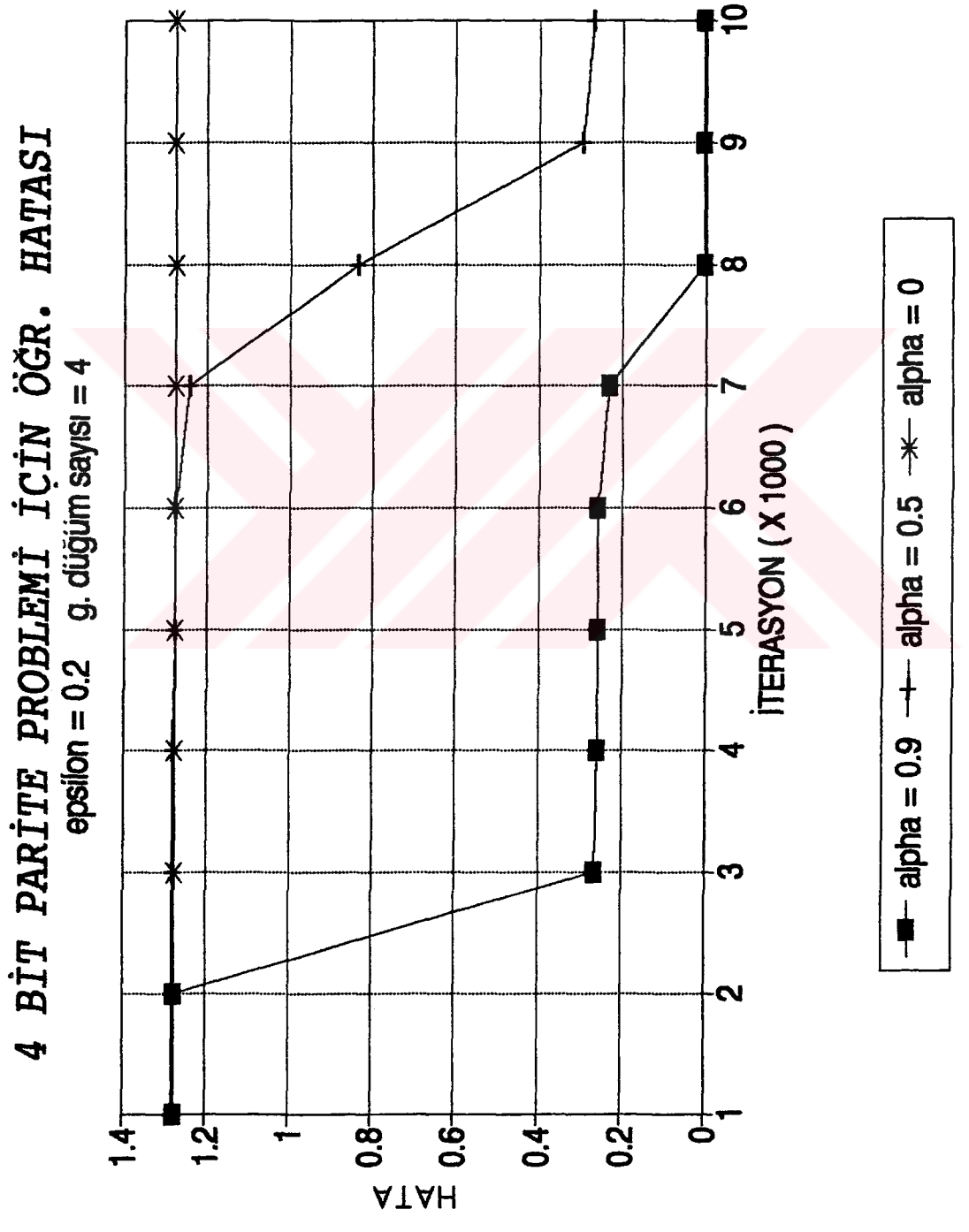
	giriş			çıkış	hedef değer
0	0.100	0.100	0.100	0.136	0.100
1	0.100	0.100	0.900	0.810	0.900
2	0.100	0.900	0.100	0.898	0.900
3	0.100	0.900	0.900	0.105	0.100
4	0.900	0.100	0.100	0.809	0.900
5	0.900	0.100	0.900	0.268	0.100
6	0.900	0.900	0.100	0.103	0.100
7	0.900	0.900	0.900	0.868	0.900

iterasyon=10000 eps= 0.077100 hata_1= 0.023778
hata_t= 0.023501

epsilon = 0.2
alpha = 0
gizli düğüm sayısı = 3

	giriş			çıkış	hedef değer
0	0.100	0.100	0.100	0.500	0.100
1	0.100	0.100	0.900	0.500	0.900
2	0.100	0.900	0.100	0.501	0.900
3	0.100	0.900	0.900	0.501	0.100
4	0.900	0.100	0.100	0.499	0.900
5	0.900	0.100	0.900	0.499	0.100
6	0.900	0.900	0.100	0.500	0.100
7	0.900	0.900	0.900	0.500	0.900

iterasyon=10000 eps= 0.405184 hata_1= 0.656695
hata_t= 0.639970


4 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
epsilon = 0.2 g. düğüm sayısı = 4
HATA
1.4
1.2
1
0.8
0.6
0.4
0.2
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
alpha = 0.9
alpha = 0.5
alpha = 0

epsilon = 0.2
alpha = 0.5
gizli düğüm sayısı = 4

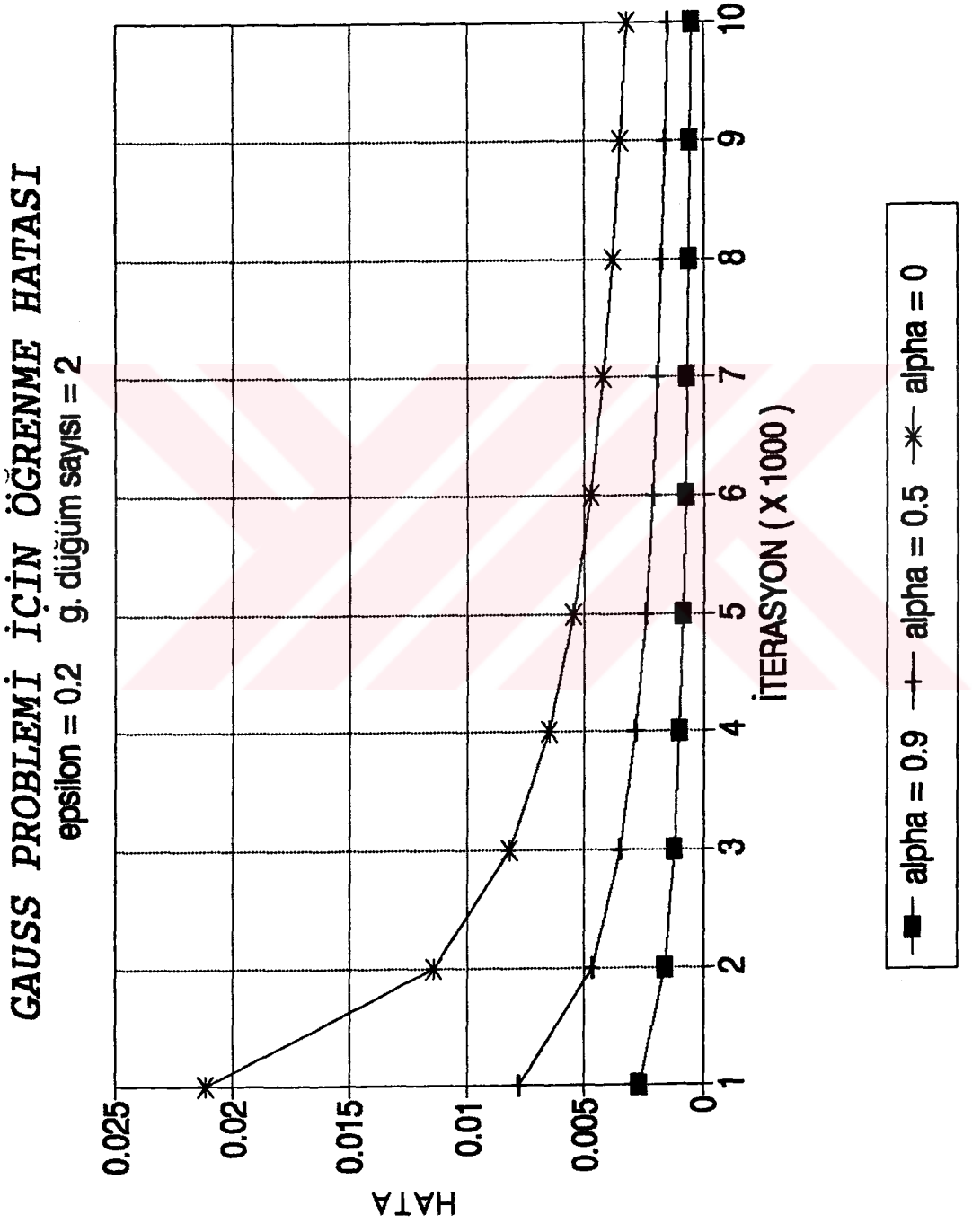
	giriş				çıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.102	0.100
1	0.100	0.100	0.100	0.900	0.897	0.900
2	0.100	0.100	0.900	0.100	0.897	0.900
3	0.100	0.100	0.900	0.900	0.118	0.100
4	0.100	0.900	0.100	0.100	0.898	0.900
5	0.100	0.900	0.100	0.900	0.118	0.100
6	0.100	0.900	0.900	0.100	0.118	0.100
7	0.100	0.900	0.900	0.900	0.738	0.900
8	0.900	0.100	0.100	0.100	0.897	0.900
9	0.900	0.100	0.100	0.900	0.116	0.100
10	0.900	0.100	0.900	0.100	0.117	0.100
11	0.900	0.100	0.900	0.900	0.738	0.900
12	0.900	0.900	0.100	0.100	0.116	0.100
13	0.900	0.900	0.100	0.900	0.738	0.900
14	0.900	0.900	0.900	0.100	0.737	0.900
15	0.900	0.900	0.900	0.900	0.759	0.100

iterasyon=10000 eps= 0.261970 hata_1= 0.274513 hata_t= 0.270830

epsilon = 0.2
alpha = 0
gizli düğüm sayısı = 4

	giriş				çıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.495	0.100
1	0.100	0.100	0.100	0.900	0.499	0.900
2	0.100	0.100	0.900	0.100	0.498	0.900
3	0.100	0.100	0.900	0.900	0.500	0.100
4	0.100	0.900	0.100	0.100	0.498	0.900
5	0.100	0.900	0.100	0.900	0.500	0.100
6	0.100	0.900	0.900	0.100	0.500	0.100
7	0.100	0.900	0.900	0.900	0.501	0.900
8	0.900	0.100	0.100	0.100	0.499	0.900
9	0.900	0.100	0.100	0.900	0.500	0.100
10	0.900	0.100	0.900	0.100	0.500	0.100
11	0.900	0.100	0.900	0.900	0.501	0.900
12	0.900	0.900	0.100	0.100	0.501	0.100
13	0.900	0.900	0.100	0.900	0.501	0.900
14	0.900	0.900	0.900	0.100	0.501	0.900
15	0.900	0.900	0.900	0.900	0.502	0.100

iterasyon=10000 eps= 0.575970 hata_1= 1.326965 hata_t= 1.279833

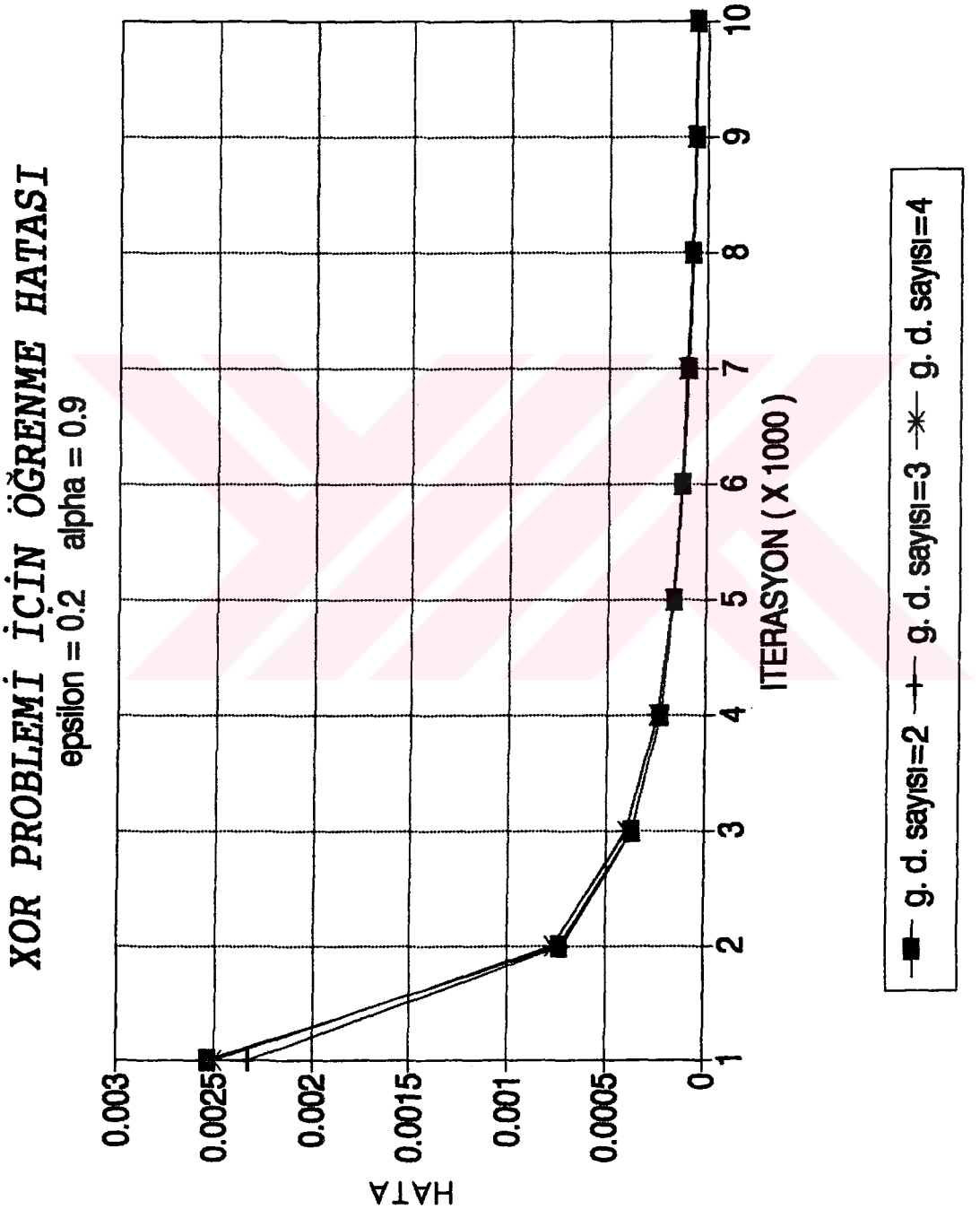

GAUSS PROBLEMİ İÇİN ÖĞRENME HATASI
epsilon = 0.2 g. düğüm sayısı = 2
HATA
0.025
0.02
0.015
0.01
0.005
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
alpha = 0.9
alpha = 0.5
alpha = 0

```
epsilon             = 0.2
alpha               = 0.5
gizli düğüm sayısı = 2

       giriş  cıkış     hedef değer
-----  -----  ------    -------------
    0  0.799  0.980         1.000
    1  -2.64  0.017         0.000
    2  -0.73  0.023         0.000
    3  -1.83  0.017         0.000
    4  -1.03  0.019         0.000
    5  1.432  0.987         1.000
    6  1.992  0.987         1.000
    7  1.729  0.987         1.000
    8  -0.88  0.020         0.000
    9  2.284  0.987         1.000
iterasyon=10000     eps= 0.019321     hata_1= 0.001493
                                      hata_t= 0.001493
```

```
epsilon             = 0.2
alpha               = 0
gizli düğüm sayısı = 2

       giriş  cıkış     hedef değer
-----  -----  ------    -------------
    0  0.555  0.966         1.000
    1  -3.21  0.017         0.000
    2  0.160  0.054         0.000
    3  0.584  0.973         1.000
    4  -1.10  0.017         0.000
    5  0.611  0.978         1.000
    6  -0.25  0.019         0.000
    7  3.145  0.991         1.000
    8  2.339  0.991         1.000
    9  4.857  0.991         1.000
iterasyon=10000     eps= 0.028363     hata_1= 0.003218
                                      hata_t= 0.003217
```

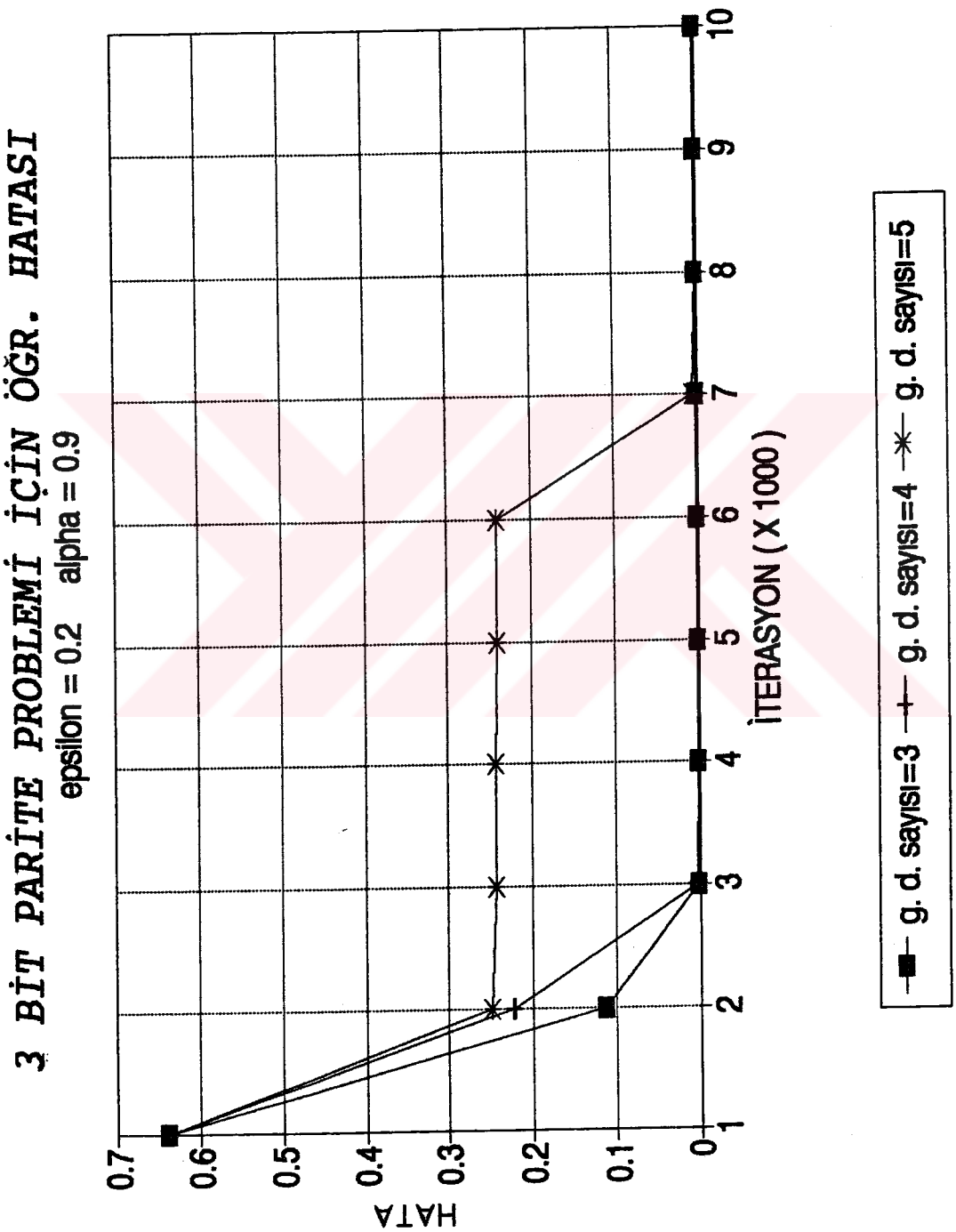
2.3 GİZLİ KATMAN DÜĞÜM SAYISI

Nöron ağı için uygun bir gizli düğüm sayısı belirlenmelidir. Gizli katmandaki düğüm sayısı gereğinden fazla artırılırsa hatanın yakınsamasının gecikmesi nedeni ile hesaplama zamanının arttığı gözlenmiştir. Gizli düğüm sayısı değiştirilirken öğrenme oranı 0.2, momentum katsayısı 0.9 alınmıştır.


XOR PROBLEMİ İÇİN ÖĞRENME HATASI
epsilon = 0.2 alpha = 0.9
HATA
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
1
2
3
4
5
6
7
8
9
10
ITERASYON (X 1000)
g. d. sayısı=2
g. d. sayısı=3
g. d. sayısı=4

```
epsilon             = 0.2
alpha               = 0.9
gizli düğüm sayısı = 3

                giriş           cıkış     hedef değer
----------  ---------------   --------  ---------------
    0        0.100   0.100      0.105       0.100
    1        0.100   0.900      0.896       0.900
    2        0.900   0.100      0.896       0.900
    3        0.900   0.900      0.105       0.100
iterasyon=10000     eps= 0.003422     hata_1= 0.000047
                                      hata_t= 0.000047
```

```
epsilon             = 0.2
alpha               = 0.9
gizli düğüm sayısı = 4

                giriş           cıkış     hedef değer
----------  ---------------   --------  ---------------
    0        0.100   0.100      0.105       0.100
    1        0.100   0.900      0.895       0.900
    2        0.900   0.100      0.895       0.900
    3        0.900   0.900      0.105       0.100
iterasyon=10000     eps= 0.003509     hata_1= 0.000049
                                      hata_t= 0.000049
```


3 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
epsilon = 0.2 alpha = 0.9
HATA
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
g. d. sayısı=3
g. d. sayısı=4
g. d. sayısı=5

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 4

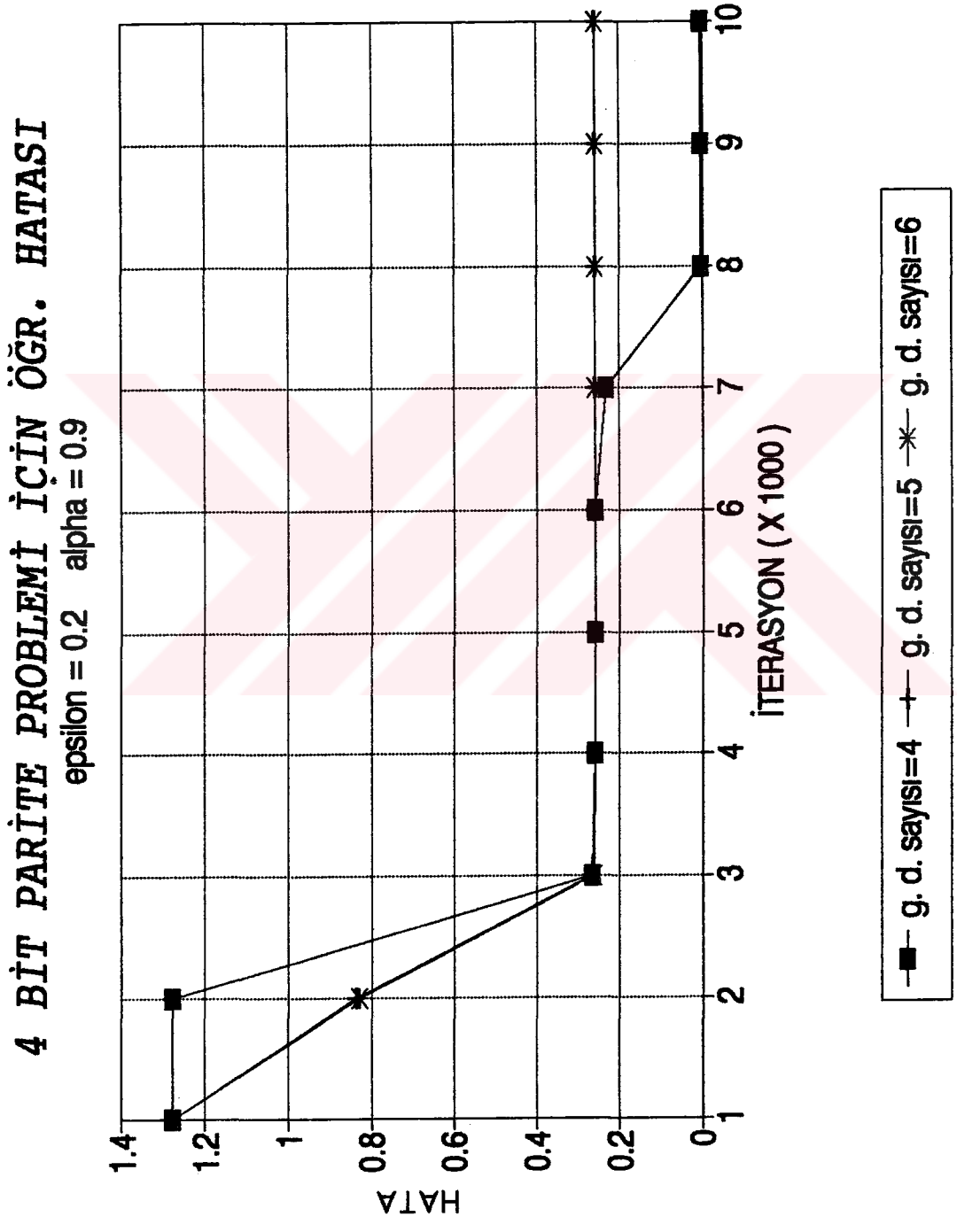
	giriş			cıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.100
1	0.100	0.100	0.900	0.899	0.900
2	0.100	0.900	0.100	0.900	0.900
3	0.100	0.900	0.900	0.104	0.100
4	0.900	0.100	0.100	0.899	0.900
5	0.900	0.100	0.900	0.104	0.100
6	0.900	0.900	0.100	0.104	0.100
7	0.900	0.900	0.900	0.889	0.900

iterasyon=10000 eps= 0.004522 hata_1= 0.000082
hata_t= 0.000082

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 5

	giriş			cıkış	hedef değer
0	0.100	0.100	0.100	0.105	0.100
1	0.100	0.100	0.900	0.898	0.900
2	0.100	0.900	0.100	0.898	0.900
3	0.100	0.900	0.900	0.107	0.100
4	0.900	0.100	0.100	0.896	0.900
5	0.900	0.100	0.900	0.107	0.100
6	0.900	0.900	0.100	0.107	0.100
7	0.900	0.900	0.900	0.880	0.900

iterasyon=10000 eps= 0.008621 hata_1= 0.000297
hata_t= 0.000297


4 BİT PARİTE PROBLEMİ İÇİN ÖĞR. HATASI
epsilon = 0.2 alpha = 0.9
HATA
İTERASYON (X 1000)
g. d. sayısı=4
g. d. sayısı=5
g. d. sayısı=6

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 5

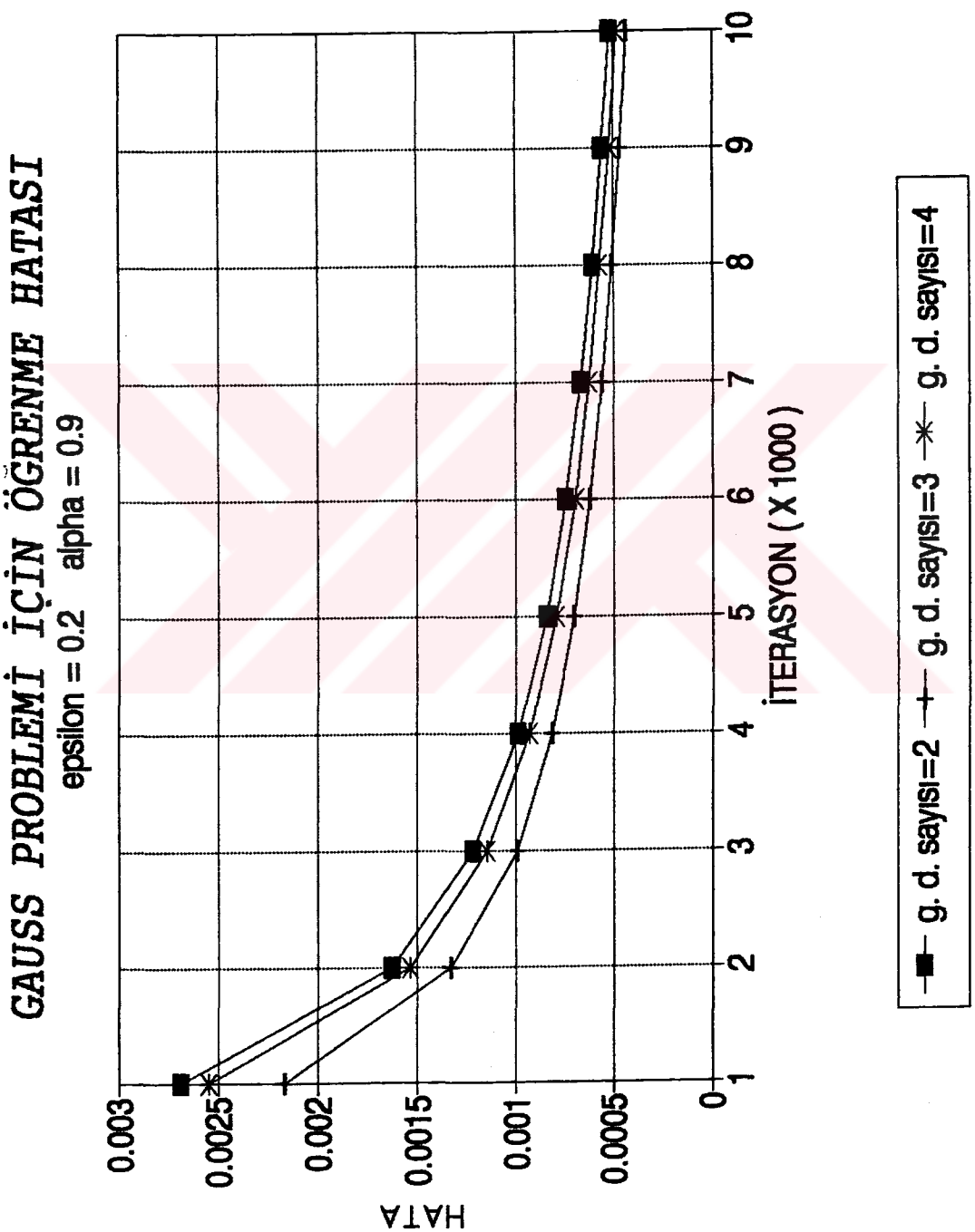
	giriş				çıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.100	0.100
1	0.100	0.100	0.100	0.900	0.901	0.900
2	0.100	0.100	0.900	0.100	0.902	0.900
3	0.100	0.100	0.900	0.900	0.104	0.100
4	0.100	0.900	0.100	0.100	0.903	0.900
5	0.100	0.900	0.100	0.900	0.104	0.100
6	0.100	0.900	0.900	0.100	0.104	0.100
7	0.100	0.900	0.900	0.900	0.747	0.900
8	0.900	0.100	0.100	0.100	0.905	0.900
9	0.900	0.100	0.100	0.900	0.104	0.100
10	0.900	0.100	0.900	0.100	0.105	0.100
11	0.900	0.100	0.900	0.900	0.747	0.900
12	0.900	0.900	0.100	0.100	0.105	0.100
13	0.900	0.900	0.100	0.900	0.747	0.900
14	0.900	0.900	0.900	0.100	0.747	0.900
15	0.900	0.900	0.900	0.900	0.748	0.100

iterasyon=10000 eps= 0.255996 hata_1= 0.262136 hata_t= 0.256742

epsilon = 0.2
alpha = 0.9
gizli düğüm sayısı = 6

	giriş				çıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.100	0.100
1	0.100	0.100	0.100	0.900	0.901	0.900
2	0.100	0.100	0.900	0.100	0.902	0.900
3	0.100	0.100	0.900	0.900	0.104	0.100
4	0.100	0.900	0.100	0.100	0.903	0.900
5	0.100	0.900	0.100	0.900	0.104	0.100
6	0.100	0.900	0.900	0.100	0.104	0.100
7	0.100	0.900	0.900	0.900	0.747	0.900
8	0.900	0.100	0.100	0.100	0.905	0.900
9	0.900	0.100	0.100	0.900	0.104	0.100
10	0.900	0.100	0.900	0.100	0.105	0.100
11	0.900	0.100	0.900	0.900	0.747	0.900
12	0.900	0.900	0.100	0.100	0.105	0.100
13	0.900	0.900	0.100	0.900	0.747	0.900
14	0.900	0.900	0.900	0.100	0.747	0.900
15	0.900	0.900	0.900	0.900	0.748	0.100

iterasyon=10000 eps= 0.255959 hata_1= 0.262060 hata_t= 0.256665


GAUSS PROBLEMİ İÇİN ÖĞRENME HATASI
epsilon = 0.2 alpha = 0.9
HATA
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
1
2
3
4
5
6
7
8
9
10
İTERASYON (X 1000)
g. d. sayısı=2
g. d. sayısı=3
g. d. sayısı=4

```
epsilon             = 0.2
alpha               = 0.9
gizli düğüm sayısı = 3

       giriş  cıkış     hedef değer
-----  -----  ------   -------------
    0  -2.77  0.007         0.000
    1  -2.03  0.007         0.000
    2  -3.69  0.007         0.000
    3  -0.29  0.009         0.000
    4  -1.75  0.007         0.000
    5  0.304  0.018         0.000
    6  4.969  0.994         1.000
    7  2.062  0.989         1.000
    8  2.162  0.990         1.000
    9  2.867  0.993         1.000
iterasyon=10000     eps= 0.010366     hata_1= 0.000430
                                      hata_t= 0.000430
```

```
epsilon             = 0.2
alpha               = 0.9
gizli düğüm sayısı = 4

       giriş  cıkış     hedef değer
-----  -----  ------   -------------
    0  0.113  0.020         0.000
    1  -1.71  0.005         0.000
    2  -0.10  0.009         0.000
    3  -0.56  0.006         0.000
    4  -0.10  0.009         0.000
    5  0.717  0.983         1.000
    6  1.475  0.996         1.000
    7  2.759  0.997         1.000
    8  1.859  0.996         1.000
    9  2.937  0.997         1.000
iterasyon=10000     eps= 0.010974     hata_1= 0.000482
                                      hata_t= 0.000482
```

BÖLÜM 3

ÇOK KATMANLI ALGILAYICININ AĞIRLIKLARA GÖRE DUYARLIĞI

3.1 DUYARLIK FONKSIYONLARI

Sıcaklık, nem, vs. gibi çevre koşulları nedeni ile sistem parametrelerinde değişme olur. Bu değişmeler sistemi temsil eden büyüklüklere etki eder.Meydana gelen değişimlerin belli sınırlar içinde kalması gerekir. Bu nedenle bu değişimlerin bulunması gereklidir.

Bir sistemde x parametresine bağlı bir büyüklük y olmak üzere x'de meydana gelen Δx değişmesi nedeni ile y'de meydana gelen değişmeyi bulmak için tanımlanan ve S[y,x] ile gösterilen duyarlık fonksiyonu dört ayrı biçimde tanımlanır.

1. Normalleştirilmiş duyarlık:

$$\frac{\Delta y}{y} = S[y,x]\frac{\Delta x}{x} \qquad S[y.x] = \frac{\partial lny}{\partial lnx} \qquad [3.1.1]$$

2. y'ye göre yarı-normalleştirilmiş duyarlık:

$$\Delta y = S[y,x]\frac{\Delta x}{x} \qquad S[y,x] = \frac{\partial y}{\partial lnx} \qquad [3.1.2]$$

3. x'e göre yarı-normalleştirilmiiş duyarlık:

$$\frac{\Delta y}{y} = S[y,x]\Delta x \qquad S[y,x] = \frac{\partial lny}{\partial x} \qquad [3.1.3]$$

4. Normalleştirilmemiş duyarlık:

$$\Delta y = S[y,x]\Delta x \qquad S[y,x] = \frac{\partial y}{\partial x} \qquad [3.1.4]$$

Yapay nöron ağında w (ağırlık) parametresine göre ağ çıkışı y(w) ile gösterilsin. w 'da meydana gelen küçük bir Δw değişmesi nedeniyle çıkışta meydana gelen değişmeyi bulmak için tanımlanan ve S[y,w] ya da S_w^y ile gösterilen duyarlık fonksiyonu aşağıdaki biçimde tanımlanmıştır.

$$S_{w_i}^{y} = \frac{\partial y}{\partial w_i} \qquad i = 1,2,.. \qquad [3.1.5]$$

W, w ağırlıklarından meydana gelen bir ağırlık vektörü olmak üzere çıkışın W' ya göre duyarlığı

$$S_{W}^{y} = [S_{w_1}^{y} \; S_{w_2}^{y} \; S_{w_3}^{y} \; .....S_{w_k}^{y}]^{t} \qquad [3.1.6]$$

ile verilir [7].

İncelenen tüm problemlerde w(k,i,j), k. katmandaki i.nörona k-1. katmandaki j. nörondan gelen ağırlığı; O[k,i] ise k. katmandaki i. nöron çıkışını temsil eder.

Xor ve parite problemlerinde k=0 giriş katmanı, k=1 gizli katman, k=2 çıkış katmanıdır. Gauss probleminde ise farklı olarak k=1 ve k=2 gizli katman, k=3 çıkış katmanıdır.

Ele alınan problemler için verilen sonuçlarda d_{ij}, i. giriş paternine ait w_j ağırlığına göre hesaplanmış duyarlıktır.

Sınıflandırma problemlerinde çok katmanlı algılayıcının duyarlık ölçümü her giriş paterni için bulunan duyarlığın maksimumu tarafından belirlenir [3].

3.2 XOR PROBLEMİ İÇİN DUYARLIK

Sistem ağırlık ve çıkışları aşağıdaki gibi ifade edilebilir.
Matrisel olarak,

$$W(1) = \begin{bmatrix} w(1,0,0) & w(1,0,1) \\ w(1,1,0) & w(1,1,1) \end{bmatrix}$$

$$W(2) = [\, w(2,0,0) \quad w(2,0,1) \,]$$

$$w_1 = w[2,0,0] \qquad w_2 = w[2,0,1]$$

$$w_3 = w[1,0,0] \qquad w_4 = w[1,0,1]$$

$$w_5 = w[1,1,0] \qquad w_6 = w[1,1,1]$$

$$O[2,0] = y_{2,0} = y$$

$$O[1,0] = y_{1,0} \qquad O[1,1] = y_{1,1}$$

$$O[0,0] = y_{0,0} \qquad O[0,1] = y_{0,1}$$

$$y_{2,0} = \frac{1}{1+e^{-(w_1*y_{1,0}+w_2*y_{1,1})}} \qquad [3.2.1]$$

$$y_{1,0} = \frac{1}{1+e^{-(w_3*y_{0,0}+w_4*y_{0,1})}} \qquad [3.2.2]$$

$$y_{1,1} = \frac{1}{1+e^{-(w_5*y_{0,0}+w_6*y_{0,1})}} \qquad [3.2.3]$$

Gizli katman çıkışlarının ağırlıklara göre türevleri:

$$\frac{\partial y_{1,0}}{\partial w_3} = y_{0,0}*y_{1,0}*(1-y_{1,0}) \qquad [3.2.4]$$

$$\frac{\partial y_{1,0}}{\partial w_4} = y_{0,1}*y_{1,0}*(1-y_{1,0}) \qquad [3.2.5]$$

$$\frac{\partial y_{1,1}}{\partial w_5} = y_{0,0} * y_{1,1} * (1-y_{1,1}) \qquad [3.2.6]$$

$$\frac{\partial y_{1,1}}{\partial w_6} = y_{0,1} * y_{1,1} * (1-y_{1,1}) \qquad [3.2.7]$$

Çıkışın ağırlıklara göre duyarlıkları aşağıdaki biçimde bulunabilir.

$$S^{y}_{w_1} = y_{1,0} * y_{2,0} * (1-y_{2,0}) \qquad [3.2.8]$$

$$S^{y}_{w_2} = y_{1,1} * y_{2,0} * (1-y_{2,0}) \qquad [3.2.9]$$

$$S^{y}_{w_3} = w_1 * y_{0,0} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.2.10]$$

$$S^{y}_{w_4} = w_1 * y_{0,1} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.2.11]$$

$$S^{y}_{w_5} = w_2 * y_{0,0} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.2.12]$$

$$S^{y}_{w_6} = w_2 * y_{0,1} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.2.13]$$

	giriş		cıkış	hedef değer
0	0.100	0.100	0.105	0.100

d0,1 = 0.000915686093
d0,2 = 0.013690672753
d0,3 = -0.000666471786
d0,4 = -0.000666471786
d0,5 = 0.007901695456
d0,6 = 0.007901695456

1	0.100	0.900	0.896	0.900

d1,1 = 0.015255631647
d1,2 = 0.088715763586
d1,3 = -0.009385031102
d1,4 = -0.084465279919
d1,5 = 0.003112381336
d1,6 = 0.028011432023

2	0.900	0.100	0.896	0.900

d2,1 = 0.015244834401
d2,2 = 0.088705195097
d2,3 = -0.084417488886
d2,4 = -0.009379720987
d2,5 = 0.028080028336
d2,6 = 0.003120003148

3	0.900	0.900	0.105	0.100

d3,1 = 0.074821613636
d3,2 = 0.094240796639
d3,3 = -0.102192361194
d3,4 = -0.102192361194
d3,5 = 0.000294502050
d3,6 = 0.000294502050

Tablo 3.2

Xor problemi için sistem duyarlıkları

ağırlık	duyarlık
w[2,0,0]	d3,1
w[2,0,1]	d3,2
w[1,0,0]	d3,3
w[1,0,1]	d3,4
w[1,1,0]	d2,5
w[1,1,1]	d1,6

3.3 3 BİT PARİTE PROBLEMİ İÇİN DUYARLIK

Sistem ağırlık ve çıkışları:

Matrisel olarak,

$$W(2) = [\, w(2,0,0) \;\; w(2,0,1) \;\; w(2,0,2) \,]$$

$$W(1) = \begin{bmatrix} w(1,0,0) & w(1,0,1) & w(1,0,2) \\ w(1,1,0) & w(1,1,1) & w(1,1,2) \\ w(1,2,0) & w(1,2,1) & w(1,2,2) \end{bmatrix}$$

$$w_1 = w[2,0,0] \quad w_2 = w[2,0,1] \quad w_3 = w[2,0,2]$$

$$w_4 = w[1,0,0] \quad w_5 = w[1,0,1] \quad w_6 = w[1,0,2]$$

$$w_7 = w[1,1,0] \quad w_8 = w[1,1,1] \quad w_9 = w[1,1,2]$$

$$w_{10} = w[1,2,0] \quad w_{11} = w[1,2,1] \quad w_{12} = w[1,2,2]$$

$$O[2,0] = y_{2,0} = y$$

$$O[1,0] = y_{1,0} \quad O[1,1] = y_{1,1} \quad O[1,2] = y_{1,2}$$

$$O[0,0] = y_{0,0} \quad O[0,1] = y_{0,1} \quad O[0,2] = y_{0,2}$$

$$y_{2,0} = \frac{1}{1+e^{-(w_1*y_{1,0}+w_2*y_{1,1}+w_3*y_{1,2})}} \qquad [3.3.1]$$

$$y_{1,0} = \frac{1}{1+e^{-(w_4*y_{0,0}+w_5*y_{0,1}+w_6*y_{0,2})}} \qquad [3.3.2]$$

$$y_{1,1} = \frac{1}{1+e^{-(w_7*y_{0,0}+w_8*y_{0,1}+w_9*y_{0,2})}} \qquad [3.3.3]$$

$$y_{1,2} = \frac{1}{1+e^{-(w_{10}*y_{0,0}+w_{11}*y_{0,1}+w_{12}*y_{0,2})}} \qquad [3.3.4]$$

Gizli katman çıkışlarının ağırlıklara göre türevleri:

$$\frac{\partial y_{1,0}}{\partial w_4} = y_{0,0}*y_{1,0}*(1-y_{1,0}) \qquad [3.3.5]$$

$$\frac{\partial y_{1,0}}{\partial w_5} = y_{0,1} * y_{1,0} * (1-y_{1,0}) \qquad [3.3.6]$$

$$\frac{\partial y_{1,0}}{\partial w_6} = y_{0,2} * y_{1,0} * (1-y_{1,0}) \qquad [3.3.7]$$

$$\frac{\partial y_{1,1}}{\partial w_7} = y_{0,0} * y_{1,1} * (1-y_{1,1}) \qquad [3.3.8]$$

$$\frac{\partial y_{1,1}}{\partial w_8} = y_{0,1} * y_{1,1} * (1-y_{1,1}) \qquad [3.3.9]$$

$$\frac{\partial y_{1,1}}{\partial w_9} = y_{0,2} * y_{1,1} * (1-y_{1,1}) \qquad [3.3.10]$$

$$\frac{\partial y_{1,2}}{\partial w_{10}} = y_{0,0} * y_{1,2} * (1-y_{1,2}) \qquad [3.3.11]$$

$$\frac{\partial y_{1,2}}{\partial w_{11}} = y_{0,1} * y_{1,2} * (1-y_{1,2}) \qquad [3.3.12]$$

$$\frac{\partial y_{1,2}}{\partial w_{12}} = y_{0,2} * y_{1,2} * (1-y_{1,2}) \qquad [3.3.13]$$

Çıkışın ağırlıklara göre duyarlığı:

$$S^{y}_{w_1} = y_{1,0} * y_{2,0} * (1-y_{2,0}) \qquad [3.3.14]$$

$$S^{y}_{w_2} = y_{1,1} * y_{2,0} * (1-y_{2,0}) \qquad [3.3.15]$$

$$S^{y}_{w_3} = y_{1,2} * y_{2,0} * (1-y_{2,0}) \qquad [3.3.16]$$

$$S^{y}_{w_4} = w_1 * y_{0,0} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.17]$$

$$S^{y}_{w_5} = w_1 * y_{0,1} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.18]$$

$$S^{y}_{w_6} = w_1 * y_{0,2} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.19]$$

$$S^{y}_{w_7} = w_2 * y_{0,0} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.20]$$

$$S^{y}_{w_8} = w_2 * y_{0,1} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.21]$$

$$S^{y}_{w_9} = w_2 * y_{0,2} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.22]$$

$$S^{y}_{w_{10}} = w_3 * y_{0,0} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.23]$$

$$S^{y}_{w_{11}} = w_3 * y_{0,1} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.24]$$

$$S^{y}_{w_{12}} = w_3 * y_{0,2} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.3.25]$$

	giriş			çıkış	hedef değer
0	0.100	0.100	0.100	0.104	0.100

d0,1 = 0.091469453062
d0,2 = 0.001621108236
d0,3 = 0.078750576825
d0,4 = 0.001248370983
d0,5 = 0.001248370983
d0,6 = 0.001248370983
d0,7 = 0.001195274124
d0,8 = 0.001195274124
d0,9 = 0.001195274124
d0,10 = -0.008940722454
d0,11 = -0.008940722454
d0,12 = -0.008940722454

1	0.100	0.100	0.900	0.896	0.900

d1,1 = 0.070897685254
d1,2 = 0.000008839121
d1,3 = 0.001553545858
d1,4 = 0.012508257615
d1,5 = 0.012508257615
d1,6 = 0.112574318537
d1,7 = 0.000006631973
d1,8 = 0.000006631973
d1,9 = 0.000059687755
d1,10 = -0.001117676954
d1,11 = -0.001117676954
d1,12 = -0.010059092585

2	0.100	0.900	0.100	0.900	0.900

d2,1 = 0.090113230353
d2,2 = 0.065890791211
d2,3 = 0.090162509314
d2,4 = 0.000071863738
d2,5 = 0.000646773638
d2,6 = 0.000071863738
d2,7 = 0.013329707092
d2,8 = 0.119967363830
d2,9 = 0.013329707092
d2,10 = -0.000036207532
d2,11 = -0.000325867785
d2,12 = -0.000036207532

	giriş			cıkış	hedef değer
3	0.100	0.900	0.900	0.099	0.100

d3,1 = 0.087752625076
d3,2 = 0.001273936401
d3,3 = 0.075919047823
d3,4 = 0.001151683385
d3,5 = 0.010365150468
d3,6 = 0.010365150468
d3,7 = 0.000942294562
d3,8 = 0.008480651060
d3,9 = 0.008480651060
d3,10 = -0.008355846235
d3,11 = -0.075202616115
d3,12 = -0.075202616115

4	0.900	0.100	0.100	0.896	0.900

d4,1 = 0.070890609589
d4,2 = 0.000008841257
d4,3 = 0.001555304556
d4,4 = 0.112539608918
d4,5 = 0.012504400991
d4,6 = 0.012504400991
d4,7 = 0.000059702176
d4,8 = 0.000006633575
d4,9 = 0.000006633575
d4,10 = -0.010070259200
d4,11 = -0.001118917689
d4,12 = -0.001118917689

5	0.900	0.100	0.900	0.109	0.100

d5,1 = 0.015275032644
d5,2 = 0.000000048949
d5,3 = 0.000005086781
d5,4 = 0.084345313101
d5,5 = 0.009371701456
d5,6 = 0.084345313101
d5,7 = 0.000000330570
d5,8 = 0.000000036730
d5,9 = 0.000000330570
d5,10 = -0.000033490858
d5,11 = -0.000003721206
d5,12 = -0.000033490858

	giriş			cıkış	hedef değer
6	0.900	0.900	0.100	0.099	0.100

d6,1 = 0.087671841489
d6,2 = 0.001273269487
d6,3 = 0.075863835468
d6,4 = 0.010352809849
d6,5 = 0.010352809849
d6,6 = 0.001150312205
d6,7 = 0.008476162062
d6,8 = 0.008476162062
d6,9 = 0.000941795785
d6,10 = -0.075063702617
d6,11 = -0.075063702617
d6,12 = -0.008340411402

7	0.900	0.900	0.900	0.900	0.900

d7,1 = 0.068878888820
d7,2 = 0.000006953038
d7,3 = 0.001551754610
d7,4 = 0.106076912750
d7,5 = 0.106076912750
d7,6 = 0.106076912750
d7,7 = 0.000046952445
d7,8 = 0.000046952445
d7,9 = 0.000046952445
d7,10 = -0.010041045110
d7,11 = -0.010041045110
d7,12 = -0.010041045110

Tablo 3.3

3 bit parite problemi için sistem duyarlıkları

agırlık	duyarlık
w[2,0,0]	d0,1
w[2,0,1]	d2,2
w[2,0,2]	d2,3
w[1,0,0]	d4,4
w[1,0,1]	d7,5
w[1,0,2]	d1,6
w[1,1,0]	d2,7
w[1,1,1]	d2,8
w[1,1,2]	d2,9
w[1,2,0]	d6,10
w[1,2,1]	d3,11
w[1,2,2]	d3,12

3.4 4 BİT PARİTE PROBLEMİ İÇİN DUYARLIK

Sistem ağırlıkları ve çıkışları:

Matrisel olarak,

$$W(2) = [\, w(2,0,0) \;\; w(2,0,1) \;\; w(2,0,2) \;\; w(2,0,3) \,]$$

$$W(1) = \begin{bmatrix} w(1,0,0) & w(1,0,1) & w(1,0,2) & w(1,0,3) \\ w(1,1,0) & w(1,1,1) & w(1,1,2) & w(1,1,3) \\ w(1,2,0) & w(1,2,1) & w(1,2,2) & w(1,2,3) \\ w(1,3,0) & w(1,3,1) & w(1,3,2) & w(1,3,3) \end{bmatrix}$$

$$w_1 = w[2,0,0] \quad w_2 = w[2,0,1] \quad w_3 = w[2,0,2] \quad w_4 = w[2,0,3]$$

$$w_5 = w[1,0,0] \quad w_6 = w[1,0,1] \quad w_7 = w[1,0,2] \quad w_8 = w[1,0,3]$$

$$w_9 = w[1,1,0] \quad w_{10} = w[1,1,1] \quad w_{11} = w[1,1,2] \quad w_{12} = w[1,1,3]$$

$$w_{13} = w[1,2,0] \quad w_{14} = w[1,2,1] \quad w_{15} = w[1,2,2] \quad w_{16} = w[1,2,3]$$

$$w_{17} = w[1,3,0] \quad w_{18} = w[1,3,1] \quad w_{19} = w[1,3,2] \quad w_{20} = w[1,3,3]$$

$$O[2,0] = y_{2,0} = y$$

$$O[1,0] = y_{1,0} \quad O[1,1] = y_{1,1} \quad O[1,2] = y_{1,2} \quad O[1,3] = y_{1,3}$$

$$O[0,0] = y_{0,0} \quad O[0,1] = y_{0,1} \quad O[0,2] = y_{0,2} \quad O[0,3] = y_{0,3}$$

$$y_{2,0} = \frac{1}{1+e^{-(w_1 * y_{1,0} + w_2 * y_{1,1} + w_3 * y_{1,2} + w_4 * y_{1,3})}} \qquad [3.4.1]$$

$$y_{1,0} = \frac{1}{1+e^{-(w_5 * y_{0,0} + w_6 * y_{0,1} + w_7 * y_{0,2} + w_8 * y_{0,3})}} \qquad [3.4.2]$$

$$y_{1,1} = \frac{1}{1+e^{-(w_9 * y_{0,0} + w_{10} * y_{0,1} + w_{11} * y_{0,2} + w_{12} * y_{0,3})}} \qquad [3.4.3]$$

$$y_{1,2} = \frac{1}{1+e^{-(w_{13}*y_{0,0}+w_{14}*y_{0,1}+w_{15}*y_{0,2}+w_{16}*y_{0,3})}} \qquad [3.4.4]$$

$$y_{1,3} = \frac{1}{1+e^{-(w_{17}*y_{0,0}+w_{18}*y_{0,1}+w_{19}*y_{0,2}+w_{20}*y_{0,3})}} \qquad [3.4.5]$$

Gizli katman çıkışlarının ağırlıklara göre türevleri:

$$\frac{\partial y_{1,0}}{\partial w_5} = y_{0,0}*y_{1,0}*(1-y_{1,0}) \qquad [3.4.6]$$

$$\frac{\partial y_{1,0}}{\partial w_6} = y_{0,1}*y_{1,0}*(1-y_{1,0}) \qquad [3.4.7]$$

$$\frac{\partial y_{1,0}}{\partial w_7} = y_{0,2}*y_{1,0}*(1-y_{1,0}) \qquad [3.4.8]$$

$$\frac{\partial y_{1,0}}{\partial w_8} = y_{0,3}*y_{1,0}*(1-y_{1,0}) \qquad [3.4.9]$$

$$\frac{\partial y_{1,1}}{\partial w_9} = y_{0,0}*y_{1,1}*(1-y_{1,1}) \qquad [3.4.10]$$

$$\frac{\partial y_{1,1}}{\partial w_{10}} = y_{0,1}*y_{1,1}*(1-y_{1,1}) \qquad [3.4.11]$$

$$\frac{\partial y_{1,1}}{\partial w_{11}} = y_{0,2}*y_{1,1}*(1-y_{1,1}) \qquad [3.4.12]$$

$$\frac{\partial y_{1,1}}{\partial w_{12}} = y_{0,3}*y_{1,1}*(1-y_{1,1}) \qquad [3.4.13]$$

$$\frac{\partial y_{1,2}}{\partial w_{13}} = y_{0,0}*y_{1,2}*(1-y_{1,2}) \qquad [3.4.14]$$

$$\frac{\partial y_{1,2}}{\partial w_{14}} = y_{0,1} * y_{1,2} * (1-y_{1,2}) \qquad [3.4.15]$$

$$\frac{\partial y_{1,2}}{\partial w_{15}} = y_{0,2} * y_{1,2} * (1-y_{1,2}) \qquad [3.4.16]$$

$$\frac{\partial y_{1,2}}{\partial w_{16}} = y_{0,3} * y_{1,2} * (1-y_{1,2}) \qquad [3.4.17]$$

$$\frac{\partial y_{1,3}}{\partial w_{17}} = y_{0,0} * y_{1,3} * (1-y_{1,3}) \qquad [3.4.18]$$

$$\frac{\partial y_{1,3}}{\partial w_{18}} = y_{0,1} * y_{1,3} * (1-y_{1,3}) \qquad [3.4.19]$$

$$\frac{\partial y_{1,3}}{\partial w_{19}} = y_{0,2} * y_{1,3} * (1-y_{1,3}) \qquad [3.4.20]$$

$$\frac{\partial y_{1,3}}{\partial w_{20}} = y_{0,3} * y_{1,3} * (1-y_{1,3}) \qquad [3.4.21]$$

Çıkışın ağırlıklara göre duyarlığı

$$S^{y}_{w_1} = y_{1,0} * y_{2,0} * (1-y_{2,0}) \qquad [3.4.22]$$

$$S^{y}_{w_2} = y_{1,1} * y_{2,0} * (1-y_{2,0}) \qquad [3.4.23]$$

$$S^{y}_{w_3} = y_{1,2} * y_{2,0} * (1-y_{2,0}) \qquad [3.4.24]$$

$$S^{y}_{w_4} = y_{1,3} * y_{2,0} * (1-y_{2,0}) \qquad [3.4.25]$$

$$S^{y}_{w_5} = w_1 * y_{0,0} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.26]$$

$$S^{y}_{w_6} = w_1 * y_{0,1} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.27]$$

$$S^{y}_{w_7} = w_1 * y_{0,2} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.28]$$

$$S^{y}_{w_8} = w_1 * y_{0,3} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.29]$$

$$S^{y}_{w_9} = w_2 * y_{0,0} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.30]$$

$$S^{y}_{w_{10}} = w_2 * y_{0,1} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.31]$$

$$S^{y}_{w_{11}} = w_2 * y_{0,2} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.32]$$

$$S^{y}_{w_{12}} = w_2 * y_{0,3} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.33]$$

$$S^{y}_{w_{13}} = w_3 * y_{0,0} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.34]$$

$$S^{y}_{w_{14}} = w_3 * y_{0,1} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.35]$$

$$S^{y}_{w_{15}} = w_3 * y_{0,2} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.36]$$

$$S^{y}_{w_{16}} = w_3 * y_{0,3} * y_{1,2} * (1-y_{1,2}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.37]$$

$$S^{y}_{w_{17}} = w_4 * y_{0,0} * y_{1,3} * (1-y_{1,3}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.38]$$

$$S^{y}_{w_{18}} = w_4 * y_{0,1} * y_{1,3} * (1-y_{1,3}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.39]$$

$$S^{y}_{w_{19}} = w_4 * y_{0,2} * y_{1,3} * (1-y_{1,3}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.40]$$

$$S^{y}_{w_{20}} = w_4 * y_{0,3} * y_{1,3} * (1-y_{1,3}) * y_{2,0} * (1-y_{2,0}) \qquad [3.4.41]$$

	giriş				çıkış	hedef değer
0	0.100	0.100	0.100	0.100	0.101	0.100

d0,1 = 0.090483451255
d0,2 = 0.090401336941
d0,3 = 0.075866431903
d0,4 = 0.090375899276
d0,5 = -0.000000155888
d0,6 = -0.000000155888
d0,7 = -0.000000155888
d0,8 = -0.000000155888
d0,9 = 0.000089325125
d0,10 = 0.000089325125
d0,11 = 0.000089325125
d0,12 = 0.000089325125
d0,13 = -0.011659672200
d0,14 = -0.011659672200
d0,15 = -0.011659672200
d0,16 = -0.011659672200
d0,17 = 0.000096904886
d0,18 = 0.000096904886
d0,19 = 0.000096904886
d0,20 = 0.000096904886

1	0.100	0.100	0.100	0.900	0.900	0.900

d1,1 = 0.089875237181
d1,2 = 0.062763840590
d1,3 = 0.002382269790
d1,4 = 0.089147076731
d1,5 = -0.000083844680
d1,6 = -0.000083844680
d1,7 = -0.000083844680
d1,8 = -0.000754602121
d1,9 = 0.020619621497
d1,10 = 0.020619621497
d1,11 = 0.020619621497
d1,12 = 0.185576593474
d1,13 = -0.002206368110
d1,14 = -0.002206368110
d1,15 = -0.002206368110
d1,16 = -0.019857312990
d1,17 = 0.000722506844
d1,18 = 0.000722506844
d1,19 = 0.000722506844
d1,20 = 0.006502561600

	giriş				cıkış	hedef değer
2	0.100	0.100	0.900	0.100	0.900	0.900

d2,1 = 0.089875647038
d2,2 = 0.062836822005
d2,3 = 0.002465296647
d2,4 = 0.089147393942
d2,5 = -0.000083842306
d2,6 = -0.000083842306
d2,7 = -0.000754580758
d2,8 = -0.000083842306
d2,9 = 0.020588409202
d2,10 = 0.020588409202
d2,11 = 0.185295682819
d2,12 = 0.020588409202
d2,13 = -0.002281100028
d2,14 = -0.002281100028
d2,15 = -0.020529900249
d2,16 = -0.002281100028
d2,17 = 0.000722586795
d2,18 = 0.000722586795
d2,19 = 0.006503281155
d2,20 = 0.000722586795

3	0.100	0.100	0.900	0.900	0.100	0.100

d3,1 = 0.060507205249
d3,2 = 0.000436487047
d3,3 = 0.000013308313
d3,4 = 0.084296268368
d3,5 = -0.020556831901
d3,6 = -0.020556831901
d3,7 = -0.185011487112
d3,8 = -0.185011487112
d3,9 = 0.000472080355
d3,10 = 0.000472080355
d3,11 = 0.004248723194
d3,12 = 0.004248723194
d3,13 = -0.000012659080
d3,14 = -0.000012659080
d3,15 = -0.000113931718
d3,16 = -0.000113931718
d3,17 = 0.004912672975
d3,18 = 0.004912672975
d3,19 = 0.044214056772
d3,20 = 0.044214056772

	giriş				cıkış	hedef değer
4	0.100	0.900	0.100	0.100	0.900	0.900

d4,1 = 0.089876307493
d4,2 = 0.062960671611
d4,3 = 0.002606258289
d4,4 = 0.089147941458
d4,5 = -0.000083836571
d4,6 = -0.000754529135
d4,7 = -0.000083836571
d4,8 = -0.000083836571
d4,9 = 0.020535130101
d4,10 = 0.184816170912
d4,11 = 0.020535130101
d4,12 = 0.020535130101
d4,13 = -0.002407644803
d4,14 = -0.021668803229
d4,15 = -0.002407644803
d4,16 = -0.002407644803
d4,17 = 0.000722681831
d4,18 = 0.006504136481
d4,19 = 0.000722681831
d4,20 = 0.000722681831

5	0.100	0.900	0.100	0.900	0.100	0.100

d5,1 = 0.060509589039
d5,2 = 0.000439335538
d5,3 = 0.000014091941
d5,4 = 0.084296798538
d5,5 = -0.020556595181
d5,6 = -0.185009356632
d5,7 = -0.020556595181
d5,8 = -0.185009356632
d5,9 = 0.000475146069
d5,10 = 0.004276314619
d5,11 = 0.000475146069
d5,12 = 0.004276314619
d5,13 = -0.000013404362
d5,14 = -0.000120639261
d5,15 = -0.000013404362
d5,16 = -0.000120639261
d5,17 = 0.004913285203
d5,18 = 0.044219566828
d5,19 = 0.004913285203
d5,20 = 0.044219566828

	giriş				cıkış	hedef değer
6	0.100	0.900	0.900	0.100	0.100	0.100

d6,1 = 0.060510983517
d6,2 = 0.000441020572
d6,3 = 0.000014596941
d6,4 = 0.084297240319
d6,5 = -0.020556614845
d6,6 = -0.185009533605
d6,7 = -0.185009533605
d6,8 = -0.020556614845
d6,9 = 0.000476959518
d6,10 = 0.004292635662
d6,11 = 0.004292635662
d6,12 = 0.000476959518
d6,13 = -0.000013884644
d6,14 = -0.000124961800
d6,15 = -0.000124961800
d6,16 = -0.000013884644
d6,17 = 0.004913807579
d6,18 = 0.044224268214
d6,19 = 0.044224268214
d6,20 = 0.004913807579

7	0.100	0.900	0.900	0.900	0.894	0.900

d7,1 = 0.000356959115
d7,2 = 0.000000981710
d7,3 = 0.000000080697
d7,4 = 0.062245098497
d7,5 = -0.000367625539
d7,6 = -0.003308629855
d7,7 = -0.003308629855
d7,8 = -0.003308629855
d7,9 = 0.000001066919
d7,10 = 0.000009602274
d7,11 = 0.000009602274
d7,12 = 0.000009602274
d7,13 = -0.000000076772
d7,14 = -0.000000690946
d7,15 = -0.000000690946
d7,16 = -0.000000690946
d7,17 = 0.019353204877
d7,18 = 0.174178843891
d7,19 = 0.174178843891
d7,20 = 0.174178843891

	giriş				cıkış	hedef değer
8	0.900	0.100	0.100	0.100	0.900	0.900

d8,1 = 0.089875533796
d8,2 = 0.063028557394
d8,3 = 0.002684253016
d8,4 = 0.089147117982
d8,5 = -0.000754492364
d8,6 = -0.000083832485
d8,7 = -0.000083832485
d8,8 = -0.000083832485
d8,9 = 0.184546473094
d8,10 = 0.020505163677
d8,11 = 0.020505163677
d8,12 = 0.020505163677
d8,13 = -0.022297329805
d8,14 = -0.002477481089
d8,15 = -0.002477481089
d8,16 = -0.002477481089
d8,17 = 0.006504500767
d8,18 = 0.000722722307
d8,19 = 0.000722722307
d8,20 = 0.000722722307

9	0.900	0.100	0.100	0.900	0.100	0.100

d9,1 = 0.060511260034
d9,2 = 0.000440929253
d9,3 = 0.000014526895
d9,4 = 0.084297654728
d9,5 = -0.185009477106
d9,6 = -0.020556608567
d9,7 = -0.020556608567
d9,8 = -0.185009477106
d9,9 = 0.004291751236
d9,10 = 0.000476861248
d9,11 = 0.000476861248
d9,12 = 0.004291751236
d9,13 = -0.000124362245
d9,14 = -0.000013818027
d9,15 = -0.000013818027
d9,16 = -0.000124362245
d9,17 = 0.044222738958
d9,18 = 0.004913637662
d9,19 = 0.004913637662
d9,20 = 0.044222738958

	giriş				çıkış	hedef değer
10	0.900	0.100	0.900	0.100	0.100	0.100

d10,1 = 0.060512610115
d10,2 = 0.000442620044
d10,3 = 0.000015047468
d10,4 = 0.084298034602
d10,5 = -0.185009518173
d10,6 = -0.020556613130
d10,7 = -0.185009518173
d10,8 = -0.020556613130
d10,9 = 0.004308127449
d10,10 = 0.000478680828
d10,11 = 0.004308127449
d10,12 = 0.000478680828
d10,13 = -0.000128818041
d10,14 = -0.000014313116
d10,15 = -0.000128818041
d10,16 = -0.000014313116
d10,17 = 0.044227408180
d10,18 = 0.004914156464
d10,19 = 0.044227408180
d10,20 = 0.004914156464

11	0.900	0.100	0.900	0.900	0.894	0.900

d11,1 = 0.000357011463
d11,2 = 0.000000985379
d11,3 = 0.000000083196
d11,4 = 0.062250321291
d11,5 = -0.003309114564
d11,6 = -0.000367679396
d11,7 = -0.003309114564
d11,8 = -0.003309114564
d11,9 = 0.000009638166
d11,10 = 0.000001070907
d11,11 = 0.000009638166
d11,12 = 0.000009638166
d11,13 = -0.000000712342
d11,14 = -0.000000079149
d11,15 = -0.000000712342
d11,16 = -0.000000712342
d11,17 = 0.174200968745
d11,18 = 0.019355663194
d11,19 = 0.174200968745
d11,20 = 0.174200968745

	giriş				cıkış	hedef değer
12	0.900	0.900	0.100	0.100	0.100	0.100

d12,1 = 0.060514702021
d12,2 = 0.000445506211
d12,3 = 0.000015933407
d12,4 = 0.084298157901
d12,5 = -0.185006494249
d12,6 = -0.185006494249
d12,7 = -0.020556277139
d12,8 = -0.020556277139
d12,9 = 0.004336079879
d12,10 = 0.004336079879
d12,11 = 0.000481786653
d12,12 = 0.000481786653
d12,13 = -0.000136401024
d12,14 = -0.000136401024
d12,15 = -0.000015155669
d12,16 = -0.000015155669
d12,17 = 0.044232706299
d12,18 = 0.044232706299
d12,19 = 0.004914745144
d12,20 = 0.004914745144

13	0.900	0.900	0.100	0.900	0.894	0.900

d13,1 = 0.000357111598
d13,2 = 0.000000992030
d13,3 = 0.000000088112
d13,4 = 0.062260358426
d13,5 = -0.003310041769
d13,6 = -0.003310041769
d13,7 = -0.000367782419
d13,8 = -0.003310041769
d13,9 = 0.000009703218
d13,10 = 0.000009703218
d13,11 = 0.000001078135
d13,12 = 0.000009703218
d13,13 = -0.000000754436
d13,14 = -0.000000754436
d13,15 = -0.000000083826
d13,16 = -0.000000754436
d13,17 = 0.174243490548
d13,18 = 0.174243490548
d13,19 = 0.019360387839
d13,20 = 0.174243490548

	giriş				cıkış	hedef değer
14	0.900	0.900	0.900	0.100	0.894	0.900

d14,1 = 0.000357185491
d14,2 = 0.000000996015
d14,3 = 0.000000091285
d14,4 = 0.062268877057
d14,5 = -0.003310726263
d14,6 = -0.003310726263
d14,7 = -0.003310726263
d14,8 = -0.000367858474
d14,9 = 0.000009742192
d14,10 = 0.000009742192
d14,11 = 0.000009742192
d14,12 = 0.000001082466
d14,13 = -0.000000781603
d14,14 = -0.000000781603
d14,15 = -0.000000781603
d14,16 = -0.000000086845
d14,17 = 0.174279662634
d14,18 = 0.174279662634
d14,19 = 0.174279662634
d14,20 = 0.019364406959

15	0.900	0.900	0.900	0.900	0.126	0.100

d15,1 = 0.000000766360
d15,2 = 0.000000002424
d15,3 = 0.000000000554
d15,4 = 0.021961597630
d15,5 = -0.000007130066
d15,6 = -0.000007130066
d15,7 = - 0.000007130066
d15,8 = -0.000007130066
d15,9 = 0.000000023706
d15,10 = 0.000000023706
d15,11 = 0.000000023706
d15,12 = 0.000000023706
d15,13 = -0.000000004746
d15,14 = -0.000000004746
d15,15 = -0.000000004746
d15,16 = -0.000000004746
d15,17 = 0.142557561394
d15,18 = 0.142557561394
d15,19 = 0.142557561394
d15,20 = 0.142557561394

Tablo3.4

4 bit parite problemi için sistem duyarlıkları

ağırlık	duyarlık
w[2,0,0]	d0,1
w[2,0,1]	d0,2
w[2,0,2]	d0,3
w[2,0,3]	d0,4
w[1,0,0]	d10,5
w[1,0,1]	d6,6
w[1,0,2]	d3,7
w[1,0,3]	d3,8
w[1,1,0]	d8,9
w[1,1,1]	d4,10
w[1,1,2]	d2,11
w[1,1,3]	d1,12
w[1,2,0]	d0,13
w[1,2,1]	d0,14
w[1,2,2]	d0,15
w[1,2,3]	d1,16
w[1,3,0]	d14,17
w[1,3,1]	d14,18
w[1,3,2]	d14,19
w[1,3,3]	d13,20

3.5 GAUSS PROBLEMİ İÇİN DUYARLIK

Sistem ağırlıkları ve çıkışları:

Matrisel olarak,

$$W(3) = [\, w(3,0,0) \;\; w(3,0,1) \,]$$

$$W(2) = \begin{bmatrix} w(2,0,0) & w(2,0,1) \\ w(2,1,0) & w(2,1,1) \end{bmatrix}$$

$$W(1) = \begin{bmatrix} w(1,0,0) \\ w(1,1,0) \end{bmatrix}$$

$$w_1 = w[3,0,0] \qquad w_2 = w[3,0,1]$$

$$w_3 = w[2,0,0] \qquad w_4 = w[2,0,1]$$

$$w_5 = w[2,1,0] \qquad w_6 = w[2,1,1]$$

$$w_7 = w[1,0,0] \qquad w_8 = w[1,1,0]$$

$$O[3,0] = y_{3,0} = y$$

$$O[2,0] = y_{2,0} \qquad O[2,1] = y_{2,1}$$

$$O[1,0] = y_{1,0} \qquad O[1,1] = y_{1,1}$$

$$O[0,0] = y_{0,0}$$

$$y_{3,0} = \frac{1}{1+e^{-(w_1 * y_{2,0} + w_2 * y_{2,1})}} \qquad [3.5.1]$$

$$y_{2,0} = \frac{1}{1+e^{-(w_3 * y_{1,0} + w_4 * y_{1,1})}} \qquad [3.5.2]$$

$$y_{2,1} = \frac{1}{1+e^{-(w_5*y_{1,0}+w_6*y_{1,1})}} \qquad [3.5.3]$$

$$y_{1,0} = \frac{1}{1+e^{-(w_7*y_{0,0})}} \qquad [3.5.4]$$

$$y_{1,1} = \frac{1}{1+e^{-(w_8*y_{0,0})}} \qquad [3.5.5]$$

Gizli katman çıkışlarının ağırlıklara göre türevleri:

$$\frac{\partial y_{2,0}}{\partial w_3} = y_{1,0}*y_{2,0}*(1-y_{2,0}) \qquad [3.5.6]$$

$$\frac{\partial y_{2,0}}{\partial w_4} = y_{1,1}*y_{2,0}*(1-y_{2,0}) \qquad [3.5.7]$$

$$\frac{\partial y_{2,1}}{\partial w_5} = y_{1,0}*y_{2,1}*(1-y_{2,1}) \qquad [3.5.8]$$

$$\frac{\partial y_{2,1}}{\partial w_6} = y_{1,1}*y_{2,1}*(1-y_{2,1}) \qquad [3.5.9]$$

$$\frac{\partial y_{1,0}}{\partial w_7} = y_{0,0}*y_{1,0}*(1-y_{1,0}) \qquad [3.5.10]$$

$$\frac{\partial y_{1,1}}{\partial w_8} = y_{0,0}*y_{1,1}*(1-y_{1,1}) \qquad [3.5.11]$$

Çıkışın ağırlıklara göre duyarlığı

$$S_{w_1}^{y} = y_{2,0}*y_{3,0}*(1-y_{3,0}) \qquad [3.5.12]$$

$$S_{w_2}^{y} = y_{2,1}*y_{3,0}*(1-y_{3,0}) \qquad [3.5.13]$$

$$S^{y}_{w_3} = w_1 * y_{1,0} * y_{2,0} * (1-y_{2,0}) * y_{3,0} * (1-y_{3,0}) \qquad [3.5.14]$$

$$S^{y}_{w_4} = w_1 * y_{1,1} * y_{2,0} * (1-y_{2,0}) * y_{3,0} * (1-y_{3,0}) \qquad [3.5.15]$$

$$S^{y}_{w_5} = w_2 * y_{1,0} * y_{2,1} * (1-y_{2,1}) * y_{3,0} * (1-y_{3,0}) \qquad [3.5.14]$$

$$S^{y}_{w_6} = w_2 * y_{1,1} * y_{2,1} * (1-y_{2,1}) * y_{3,0} * (1-y_{3,0}) \qquad [3.5.17]$$

$$S^{y}_{w_7} = [w_1 * w_3 * y_{0,0} * y_{1,0} * (1-y_{1,0}) * y_{2,0} * (1-y_{2,0}) +$$

$$w_2 * w_5 * y_{0,0} * y_{1,0} * (1-y_{1,0}) * y_{2,1} * (1-y_{2,1})] * y_{3,0} * (1-y_{3,0}) \qquad [3.5.18]$$

$$S^{y}_{w_8} = [w_1 * w_4 * y_{0,0} * y_{1,1} * (1-y_{1,1}) * y_{2,0} * (1-y_{2,0}) +$$

$$w_2 * w_6 * y_{0,0} * y_{1,1} * (1-y_{1,1}) * y_{2,1} * (1-y_{2,1})] * y_{3,0} * (1-y_{3,0}) \qquad [3.5.19]$$

	giriş	cıkış	hedef değer
0	-1.99	0.008	0.000

d0,1 = 0.000748578040
d0,2 = 0.000206208451
d0,3 = 0.000000397547
d0,4 = 0.000004067814
d0,5 = 0.000000246339
d0,6 = 0.000002520607
d0,7 = -0.000004305483
d0,8 = -0.000027855989

1	-1.04	0.008	0.000

d1,1 = 0.000783725980
d1,2 = 0.000219874248
d1,3 = 0.000007475261
d1,4 = 0.000036508199
d1,5 = 0.000004725391
d1,6 = 0.000023078193
d1,7 = -0.000042677949
d1,8 = -0.000130375856

2	0.177	0.014	0.000

d2,1 = 0.002236350694
d2,2 = 0.000918017627
d2,3 = 0.000722278026
d2,4 = 0.001267097854
d2,5 = 0.000692192019
d2,6 = 0.001214317743
d2,7 = 0.000783422409
d2,8 = 0.000780911617

3	-0.30	0.009	0.000

d3,1 = 0.001030112160
d3,2 = 0.000321887391
d3,3 = 0.000090605361
d3,4 = 0.000240646926
d3,5 = 0.000064426724
d3,6 = 0.000171116728
d3,7 = -0.000153808142
d3,8 = -0.000245368739

4	-1.88	0.008	0.000

d4,1 = 0.000749762601
d4,2 = 0.000206663586
d4,3 = 0.000000560333
d4,4 = 0.000005258054
d4,5 = 0.000000347445
d4,6 = 0.000003260357
d4,7 = -0.000005728432
d4,8 = -0.000033973723

	giriş	cıkış	hedef değer
5	2.233	0.991	1.000

d5,1 = 0.007732618908
d5,2 = 0.008372076840
d5,3 = 0.002919509401
d5,4 = 0.002855073468
d5,5 = 0.002243979599
d5,6 = 0.002194453154
d5,7 = 0.000599565984
d5,8 = 0.000900527810

6	3.225	0.992	1.000

d6,1 = 0.007302069460
d6,2 = 0.007865906069
d6,3 = 0.002564209967
d6,4 = 0.002555880292
d6,5 = 0.001849174723
d6,6 = 0.001843167795
d6,7 = 0.000036567940
d6,8 = 0.000122110900

7	3.321	0.992	1.000

d7,1 = 0.007294358006
d7,2 = 0.007856817131
d7,3 = 0.002557587561
d7,4 = 0.002550802093
d7,5 = 0.001842145048
d7,6 = 0.001837257701
d7,7 = 0.000027995974
d7,8 = 0.000100787158

8	1.596	0.987	1.000

d8,1 = 0.010666270410
d8,2 = 0.011793586874
d8,3 = 0.005238990078
d8,4 = 0.004984977231
d8,5 = 0.005281121182
d8,6 = 0.005025065606
d8,7 = 0.005736103643
d8,8 = 0.004981072557

9	0.124	0.013	0.000

d9,1 = 0.001940889388
d9,2 = 0.000760982360
d9,3 = 0.000548952797
d9,4 = 0.001008653310
d9,5 = 0.000500353009
d9,6 = 0.000919355400
d9,7 = 0.000411434386
d9,8 = 0.000432439306

Tablo 3.5

Gauss problemi için sistem duyarlıkları

ağırlık	duyarlık
w[3,0,0]	d8,1
w[3,0,1]	d8,2
w[2,0,0]	d8,3
w[2,0,1]	d8,4
w[2,1,0]	d8,5
w[2,1,1]	d8,6
w[1,0,0]	d8,7
w[1,1,0]	d8,8

BÖLÜM 4

SONUÇ

Bu çalışmada çok katmanlı algılayıcının gizli katmandaki düğüm sayısına, öğrenme oranındaki ve momentum katsayısındaki değişimlere göre performansı incelendi. Bunun için öğrenme ve test hataları referans olarak alındı. Öğrenme oranı artırıldığında xor ve gauss problemi için başlangıçta hata değerinin büyük olduğu ancak daha sonra azaldığı ; parite problemlerinde sistem daha büyüdüğü için başlangıçta hata değerinin küçüldüğü ancak daha sonraki iterasyonlarda hata yakınsamasının geciktiği gözlendi.

Momentum katsayısının azaltılması tüm problemlerde hatanın büyümesine yol açtı. Çünkü momentum katsayısı azaltıldığında bir önceki ağırlık değişşminin o andaki ağırlık değişimine etkisi azalmaktadır.

Gizli katmandaki düğüm sayısını artırmanın, basit bir ağ yapısına sahip olan xor problemi için hata değerinde pek fazla değişim yapmadığı, ancak parite problemlerinde sistem büyüdüğü için hesaplamaların artması nedeni ile yakınsama zamanının geciktiği gözlendi. Gauss problemi için ise gizli katmanda 3 düğüm kullanmanın hatayı minimize etmek açısından uygun bir çözüm olduğu bulundu.

Ayrıca çok katmanlı algılayıcının, ağırlıklara göre duyarlığı ele alındı. Duyarlık tanımına göre ağ ağırlıklarında meydana gelen küçük bir değişimin ağ çıkışını nasıl etkilediğini bulmak için çıkışın ayrı ayrı ağırlıklara göre türevi hesaplandı. Sistem çıkışının ağırlıklara göre duyarlığı formüller halinde ifade edildi. Ele alınan tüm problemler için 10000 iterasyondan sonraki ağırlık ve düğüm çıkışları kullanılarak; elde edilen formüller yardımı ile duyarlık hesabı yapıldı.

EK 1

BİLGİSAYAR PROGRAMI

```
PROGRAM Xor_Problemi;
{$N+,E+}
CONST
Layers_Max     = 3;      { Maksimum katman sayısı }
Nodes_Max      = 2;      { Her katmandaki maksimum düğüm sayısı }
Outputs_Max    = 4;      { Giris kombinasyon sayısı }
Input_Nodes    = 2;      { Giris katmanındaki düğüm sayısı }
Output_Nodes   = 1;      { Cikis katmanındaki düğüm sayısı }
Hidden_Nodes   = 2;      { Gizli katmandaki düğüm sayısı }
Epsilon        = 0.2;
Alpha           = 0.9;
Iterative_Max  = 10000;  { Maksimum iterasyon sayısı }
Every          = 1000;   { Her iterasyon sonrası sonuç yazılması }
VAR
i, ii, j, count, isylla, num : integer;
eps, alp, err, error, terr, terror, total_rate: double;
Layers, Inputs, Outputs : integer;
Node      : array [0..Layers_Max-1] of integer;
matrix    : array [0..Outputs_Max-1, 0..(Input_Nodes+Output_Nodes-1)] of double;
matrix_tmp: array [0..trunc(Iterative_Max/Every), 0..Outputs_Max-1,
                  0..(Input_Nodes+Output_Nodes-1)] of double;


des     : double;
rate    : array [0..Outputs_Max-1] of double;
out     : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
der     : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
theta   : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
dtheta  : array [0..Layers_Max-1, 0..Nodes_Max-1] of double;
weight  : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dweight : array [0..Layers_Max-1, 0..Nodes_Max-1, 0..Nodes_Max-1] of double;
dess    : array [0..Outputs_Max-1] of double;

st1: string[5];
st2, st3, st4: string[10];
outfile : text;


procedure Wrand;
{ Program başlangıcında agırlık ve düğüm ofset değerlerinin belirlenmesi }
var
li, ni, ni_end, no, no_end : integer;
drand48, power: double;
begin
  power:=1.0;
  for li:=1 to 31 do
      power:=power*2;
  power:=power-1;
  { Randomize offsets }
  for li:=1 to Layers_Max-1 do
  begin
    ni_end:=node[li];
    for ni:=0 to ni_end-1 do
    begin
      drand48:=Random(30)/100;
      theta[li,ni]:= drand48;
```

```
      dtheta[li,ni]:=0;
    end;
  end;

  { Rasgele ağırlıklar }
  for li:=1 to Layers_Max-1 do
  begin
    ni_end:=node[li];
    no_end:=node[li-1];
    for ni:=0 to ni_end-1 do
        for no:=0 to no_end-1 do
        begin
          drand48:=Random(100)/100;
          weight[li,ni,no]:=drand48;
          dweight[li,ni,no]:=0;
        end;
  end;
end;  { procedure WRAND }


procedure Initial;
{ Her iterasyondan önce her bir dugüm çıkışının sıfıra set edilmesi }
VAR
l, n, n_end: integer;
begin
  for l:= 0 to Layers-1 do
  begin
    n_end:= node[l];
    for n:= 0 to n_end-1 do
        out[l,n]:= 0;
  end;
end;    { procedure INITIAL }


procedure yread ( isylla: integer );
{ Giriş paternini oku ( dört farklı giriş kombinasyonu için ) }
begin
  case isylla of
       0: begin out[0,0]:=0.1; out[0,1]:=0.1; end;
       1: begin out[0,0]:=0.1; out[0,1]:=0.9; end;
       2: begin out[0,0]:=0.9; out[0,1]:=0.1; end;
       3: begin out[0,0]:=0.9; out[0,1]:=0.9; end;
  end;
end;  { procedure YREAD }


function sigmoid( x: double ): double;
{ sigmoid fonksiyonu }
var
y: double;
begin
  y:=1/(1+exp(-x));
  sigmoid:=y;
end;  { function SIGMOID }
```

```
procedure wforward;
{ ileri yön propagasyonu }
var
li, ni, ni_end, no, no_end: integer;
x, th: double;
begin
  for li:=1 to Layers-1 do
  begin
    ni_end:= node[li];
    no_end:= node[li-1];
    for ni:= 0 to ni_end-1 do
    begin
      x:= 0;
      for no:= 0 to no_end-1 do
          x:= x + weight[li,ni,no] * out[li-1,no];
      out[li,ni]:= sigmoid( x-theta[li,ni] );
    end;
  end;
end;   { procedure FORWARD }


function back: double;
{ geriye doğru propagasyon }
var
li, ni, ni_end, no, no_end: integer;
e, err, x, d, w, y: double;
begin
  err:=0;
  for ni:=0 to outputs-1 do
  begin
    y:= out[2,ni];
    e:= des - y;
    der[2,ni]:= e*y*(1-y);
    err:=err+ e*e;
  end;

  for li:=2 downto 1 do
  begin
    no_end:= node[li-1];
    ni_end:= node[li];
    for no:= 0 to no_end-1 do
    begin
      x:= 0;
      y:= out[li-1,no];
      for ni:= 0 to ni_end-1 do
      begin
        d:= der[li,ni];
        w:= weight[li,ni,no];
        x:= x + d*w;
      end;
      der[li-1,no]:= y*(1-y)*x;
    end;
  end;
  back:= 0.5*err;
end; { function BACK }
```

```
procedure learning (alp, eps: double );
{ agırlık ve ofset değerlerini uyarlama}
var
li, lo, ni, ni_end, no, no_end: integer;
di, yo, ew, et: double;
begin
  for li:= 1 to layers-1 do
  begin
    ni_end:= node[li];
    for ni:= 0 to ni_end-1 do
    begin
      et:= der[li,ni];
      dtheta[li,ni]:= -eps*et + alp*dtheta[li,ni];
      theta[li,ni]:= theta[li,ni] + dtheta[li,ni];
    end;
  end;
  for li:=1 to layers-1 do
  begin
    lo:= li-1;
    ni_end:= node[li];
    no_end:= node[lo];
    for ni:= 0 to ni_end-1 do
    begin
      di:= der[li,ni];
      for no:= 0 to no_end-1 do
      begin
        yo:= out[lo,no];
        ew:= di*yo;
        dweight[li,ni,no]:= eps*ew + alp*dweight[li,ni,no];
        weight[li,ni,no]:= weight[li,ni,no] + dweight[li,ni,no];
      end;
    end;
  end;
end;  { procedure LEARNING }


procedure make_matrix( it, isylla: integer );
var
i,j: integer;
begin
  for i:= 0 to inputs-1 do
      matrix_tmp[it,isylla,i]:= out[0,i];
  for i:= 0 to outputs-1 do
      matrix_tmp[it,isylla,inputs+i]:= out[Layers_Max-1,i];
end;  { procedure MAKE_MATRIX }


begin { main }


  FOR I:= 0 TO 0 DO WRITELN ('LALE');
  node[2]:= Output_Nodes;
  Outputs:= Output_Nodes;
  node[1]:= Hidden_Nodes;
  node[0]:= Input_Nodes;
```

```
  Inputs:= Input_Nodes;
  Layers:= Layers_Max;
  eps:= Epsilon;
  alp:= Alpha;
  num:= Outputs_Max;  { here 4 }



{ ************************
    HEDEF DEĞER OKUMA
  ************************ }
  dess[0]:= 0.1;
  dess[1]:= 0.9;
  dess[2]:= 0.9;
  dess[3]:= 0.1;

{ ********************** ~
    MATRİS BELİRLEME
  ********************** }
  for i:= 0 to num-1 do
      for j:= 0 to ( inputs+Outputs-1 ) do
          matrix[i,j]:= 0;

  for ii:= 0 to ( Trunc(Iterative_Max/Every)-1 ) do
      for i:= 0 to num-1 do
          for j:= 0 to ( inputs+Outputs-1 ) do
              matrix_tmp[ii,i,j]:=0;

{ öğrenme baslangıcı .... }

  Wrand;
  j:= 0;
  assign ( outfile,'OUTFILE.TXT');
  rewrite( outfile );
  { ************************************
            ÖĞRENME FAZI
  ************************************ }
  while ( j < Iterative_Max ) do
  begin
    for count:= 0 to Every-1 do
    begin
      error:= 0;
      for isylla:=0 to num-1 do
      begin
        initial;
        yread( isylla );
        des:=dess[ isylla ]; { hedef patern okuma }
        wforward;
        err:= back;
        learning( alp, eps );
        error:= error+err;
      end;
      error:= error / outputs;
      eps:= 0.5 * sqrt( error );
      Inc( j );
    end;
```

```
{ ********************************************
                TEST FAZI
  ******************************************** }
    terror:= 0;
    for isylla:= 0 to num-1 do
    begin
      yread( isylla );
      wforward;
      des:=dess[ isylla ];
      terr:= back;
      terror:= terror + terr;
      make_matrix( trunc( j/Every ) - 1, isylla );
    end;
    writeln (outfile, '                  giriş          cıkış     hedef değer');
    writeln (outfile, '-----------   ---------------   --------  --------------');
    for i:= 0 to num-1 do
    begin
      str( i:5, st1 );
      write(outfile,st1+'       ');
      for ii:= 0 to inputs-1 do
      begin
        str( matrix_tmp[trunc(j/Every)-1,i,ii]:4:3, st1 );
        write(outfile,st1+'   ');
      end;
      write( outfile,'   ');
      for ii:= 0 to outputs-1 do
      begin
        str( matrix_tmp[trunc(j/Every)-1,i,inputs+ii]:4:3, st1);
        write( outfile, st1+'      ');
      end;
      str( dess[i]:4:3, st1 );
      writeln( outfile, st1 );
    end;
    terror:= terror/outputs;
    str ( j:5, st1 ); str ( eps:9:6, st2 );
    str ( error:9:6, st3 ); str ( terror:9:6, st4 );
    writeln( outfile, 'iterasyon=',st1,'    eps=',st2,
             '    hata_i=', st3, '    hata_t=', st4 );
    writeln( outfile );
{ }
    total_rate:=0;
    for isylla:= 0 to outputs-1 do
    begin
      rate[isylla]:= trunc(matrix_tmp[trunc(j/Every)-1,isylla,isylla]);
      total_rate:= total_rate + rate[isylla];
    end;
    total_rate:= total_rate/outputs;
{ }
    if ( j = Iterative_Max ) then
    begin
      writeln( outfile, 'öğrenme ve test sona erdi' );
    end;
  end;
  close ( outfile );
  end.
```

KAYNAKLAR

[1] Parallel Distributed Processing Vol 1: Foundations MIT Press Eight Printing 1988

[2] Richard P. Lippmann, 'An Introduction To Computing With Neural Nets', IEEE Assp Magazine April 1987

[3] Jin Young Choi and Chong-Ho Choi, 'Sensitivity Analysis Of Multilayer Perceptron With Differentiable Activation Functions' IEEE Transactions On Neural Networks, Vol:3, No:1 January 1992

[4] Eric R. Buhrke and Joseph L. LoCicero, 'Fast Learning For Multilayer Perceptrons Using Statistical Techniques' IEEE ICASSP proc. 1992

[5] Robert Slaviera, 'Optimising Backpropagation Learning Performance And Evaluation Of Neural Network Implementation Using DSP's' ICSPAT proc. 1992

[6] Erkin Özmeteler, 'Yapay Nöral Ağlar' Yüksek Lisans Tezi İstanbul Teknik Üniversitesi 1989

[7] Doç. Dr. Cevdet Acar, 'Duyarlık ve Tolerens Analizi' İ.T.Ü. Elektrik Fakültesi Devreler ve Sistemler Kürsüsü İstanbul 1979

[8] Yoh-Han Pao, 'Adaptive Pattern Recognition and Neural Networks' Case Western Reserve University

[9] Doç. Dr. Fikret Gürgen, 'Ders Notları' ,Boğaziçi Üniversitesi Bilgisayar Mühendisliği Bölümü

ÖZGEÇMİŞ

29 Mayıs 1970 tarihinde İstanbul'da doğdu. İlkokulu Bartın Cumhuriyet İlkokulunda, ortaokulu Bartın Lisesinin orta kısmında, Lise 1 ve 2'yi Üsküdar Burhan Felek Lisesinde tamamladı. 1987 yılında Ataköy Lisesinden , 1991 yılında Yıldız Üniversitesi Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümünden mezun oldu. 1991 - 1992 öğretim yılında yüksek lisans öğrenimine başladı. Şu anda Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümünde Araştırma Görevlisi olarak görev yapmaktadır.