

# MLP YAPAY SİNİR AĞLARINDA ÖĞRENME SÜRECİNİN AKTİVASYON FONKSİYONU VE İSTATİKSEL DEĞİŞİM GÖSTEREN GİRİŞ VERİLERİNE BAĞIMLILIĞI

Halis ALTUN<sup>1\*</sup> Ulaş EMİNOĞLU<sup>2</sup> Bekir Sami TEZEKİCİ<sup>3</sup>  
Niğde Üniversitesi, Mühendislik-Mimarlık Fakültesi  
Elektrik-Elektronik Mühendisliği Bölümü,  
Niğde 51200 / Türkiye

<sup>1</sup>e-posta: haltun@alp.nigde.edu.tr <sup>2</sup>e-posta: ulas41@hotmail.com  
<sup>3</sup>e-posta: bstezekici@alp.nigde.edu.tr

Anahtar sözcükler: Yapay Sinir Ağı, Aktivasyon fonksiyonları, Eğitim Seti, Geri Yayılım Algoritması

## ABSTRACT

*In this work the dependency of neural learning in a Multilayered Perceptron MLP neural network NN is investigated on modified input data sets and the combination of the activation functions. For a 3-parity problem the symbols {a<sub>i</sub>} that stand for logic 0 and logic 1 are presented by different values ranging from -3 to 3, incremented by a factor of 0.2 in a regular fashion. In this way total 464 data input data sets obtained and the statistical characteristics of the input data sets such as mean value and standard deviation are modified. 4 configurations of MLP NN are constructed by using different activation functions of neurons in each layer. For each configuration 8 initial weight sets are determined and total 14848 experiments are carried out. The performance of the networks are determined from the results and conclusion are drawn.*

## 1. GİRİŞ

Yapay Sinir Ağlarında (YSA) nöral öğreniminin optimize edilmesi problemi literatürde yaygın olarak incelenmiştir. Geniş olarak kullanılan çok katmanlı perseptron (Multi-Layered Perceptron: MLP) tipi YSA'larının optimizasyonu alanında çok geniş bir literatür oluşturulmuştur. Özellikle öğrenme algoritması, eğitim veri setinin uygun kodlanması veya ağ topolojisinin optimize edilmesi üzerinde geniş olarak durulmuştur [1-5].

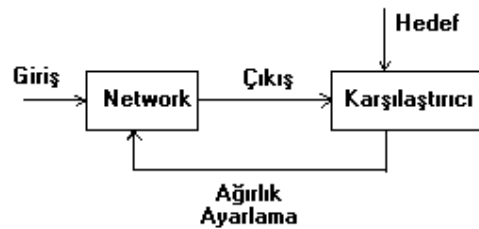
Gerçek yayılım algoritmasının hızının artırılması konusunda nöron çıkışlarını belirleyen sigmoid aktivasyon fonksiyonunun parametrelerinin değişiminin etkisi incelenmiştir [6]. Söz konusu çalışmada tüm katmanlarda sigmoid aktivasyon fonksiyonu kullanılarak bu fonksiyonun simetri, eğitim ve dinamik değişim aralığının farklı değerleri için nöral öğrenme incelenmiştir. Bu çalışmada bir MLP YSA yapısında, katmanlar arasında kullanılan farklı

aktivasyon fonksiyonlarının ve farklı kodlama kullanılarak istatistiksel özelliği değiştirilmiş giriş verilerinin eğitim performansına olan etkisi incelenmiştir. Bu amaç için katmanlar arasında üç farklı aktivasyon fonksiyonunun kombinasyonu kullanılmıştır. Giriş veri setinin istatistiksel özellikleri ise kodlamada kullanılan sembollerinin farklı sayısal değerler ile ifade edilmesi ile gerçekleştirilmiştir. Elde edilen sonuçların ilk ağırlıkların başlangıç değerlerine olan bağımlılığını azaltmak için 8 farklı noktadan eğitim başlatılmış ve ortalama sonuçları verilmiştir. Toplam olarak 14848 ağ eğitim denemesi yapılmış ve elde edilen sonuçlar tartışılmıştır.

Makalenin düzenlenmesi şu şekilde yapılmıştır. Bölüm 2 de yapay sinir ağları üzerinde genel bilgi verilmiş ve ağ yapılarında kullanılan aktivasyon fonksiyonları verilmiştir. Ayrıca ağ eğitim algoritması klasik formu ile verilmiş ve giriş veri setinin etkisini açık olarak gösterme amacı ile algoritmanın giriş, saklı katman ve çıkış nöron aktivasyon büyüklüklerine olan bağımlılığı açık olarak ifade edilmiştir. 3. bölümde yapılan deneysel çalışmalar ayrıntılı olarak vermekte ve son bölüm ise elde edilen sonuçları tartışmaktadır.

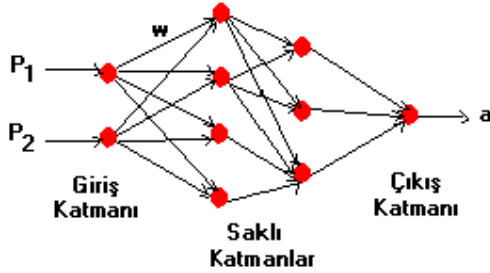
## 2. YAPAY SİNİR AĞLARI ve EĞİTİMİ

Genel olarak bir yapay sinir ağı blok diyagramı aşağıda verilmiştir.



Şekil 1. YSA Blok Diyagramı

Giriş ve ilgili çıkış vektörü ağı eğtmek için kullanılmaktadır. Her bir iterasyon sonucu elde edilen çıkış, hedef ile karşılaştırılır ve verilen hataya bağlı olarak ağırlık yenileme işlemi ile ağı eğitimi devam ettirilir veya eğitme işlemi sonlandırılır. MLP tipi çok katmanlı yapay sinir ağı giriş, saklı ve çıkış katmanlarından oluşmaktadır. Herbir katmana ait nörona toplam giriş bir alt katmandaki nöron çıkışlarının ağırlıklandırılması ile elde edilir. Nöron çıkışları ise o nöron için tanımlanmış aktivasyon fonksiyonuna bağlı olarak elde edilmektedir. Dört katmanlı bir yapay sinir ağına nöron modeli aşağıda verilmiştir.



Şekil-2 Dört katmanlı ağı yapısı

Katmanları oluşturan nöron çıkışları ağırlıklanan girişler ve bayeslerin toplamı sonucunda aşağıda verilen ifade tarafından oluşturulmaktadır.

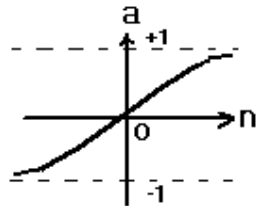
$$a = f(wp + b) \quad (1)$$

Burada  $f$  nöron için tanımlanmış aktivasyon fonksiyonu,  $w$  nöronlar arası ağırlıklar,  $p$  giriş vektörü  $b$  ise ağırlıklardır.

## 2.1 Aktivasyon Fonksiyonları

Bu çalışmada MLP tipi yapay sinir ağı için katmanlar arasında kullanılan aktivasyon fonksiyonları sırasıyla; a) Tangent-Sigmoid: tansig b)logaritmik-Sigmoid: logsig ve c) Purelin olarak adlandırılmıştır.

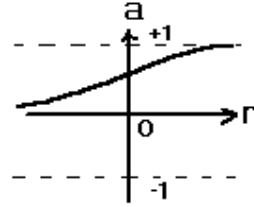
**Tansig:** Bu aktivasyon fonksiyonu için nöron giriş-çıkış ifadesi (1)'de ve fonksiyonun değişimi Şekil 2'de verilmiştir. Fonksiyonun dinamik değişim aralığı  $[-1 \ 1]$  aralığıdır ve fonksiyon nöron toplam girişe bağımlı olarak bu aralıkta lineer olmayan bir değişim gösterir. Bu fonksiyon literatürde hiperbolik-tangent fonksiyonu olarak da isimlendirilmektedir.



Şekil 2. Tangent-Sigmoid fonksiyonu giriş-çıkış eğrisi

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

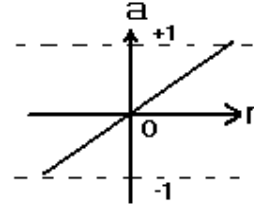
**Logsig:** Sigmoid fonksiyonu olarak da isimlendirilen bu aktivasyon fonksiyonunun giriş-çıkış ifadesi ve fonksiyonun girişe göre değişimi sırasıyla (2) ifadesinde ve Şekil 3'de verilmektedir. Fonksiyonun dinamik değişim aralığı  $[0 \ 1]$  aralığıdır ve fonksiyon bu aralıkta lineer olmayan bir değişim sergiler.



Şekil 3. Logaritmik Sigmoid fonksiyonu giriş-çıkış eğrisi

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

**Purelin:** Bu aktivasyon fonksiyonunda nöron girişlerinin değişimine göre nöron çıkışı lineer olarak değişmektedir. Dinamik değişim aralığı  $[-1 \ 1]$  aralığıdır. Fonksiyona ait giriş-çıkış karakteristiği ve fonksiyon tanımı aşağıda verilmiştir.



Şekil 4. Purelin fonksiyonu çıkış eğrisi

$$a = \text{Purelin}(n) = n \quad (3)$$

## 2.2 Geri Yayılım Algoritması

MLP tipi YSA'ların eğitiminde oldukça yaygın olarak kullanılan Geri Yayılım algoritması, giriş katman-saklı katman ve saklı katman-çıkış katmanları arasında mevcut olan ağırlıklı bağlantıların,  $w_{ih}$  ve  $w_{ho}$ , ağırlık çıkışında oluşan hataya bağlı olarak optimize edilmesini amaçlar. Giriş ve ilgili çıkış vektörü ağı eğtmek için kullanılmaktadır. Örneği belirleyen giriş vektörü ağına ilk katmanına verildikten sonra aradaki gizli katmanlardan geçerek en sondaki çıkış katmanına ağırlıklar yardımı ile ulaşır. Ağıdaki her bir nöron, kendisinde sonlanan ağırlık vektörlerinin aritmetiksel toplamını aldıktan sonra sonucu kendinden sonraki katmanın tüm nöronlarına kullanılan aktivasyon fonksiyonuna bağlı olarak aktarılır.

Katmanlar arasında sigmoid aktivasyon fonksiyonu kullanıldığı kabulü altında ağırlıkların yenilenmesi için kullanılan ifadeler şu şekilde elde edilebilir: Çıkış katmanındaki her bir nöron için nöron çıkışı

$$O_k = \frac{1}{1 + e^{-net_k}} \quad (4)$$

olsun.

Burada  $O_k$  çıkış katmanının aktivasyon değerini göstermektedir.

$$net_k = \sum_j W_{jk} O_j \quad (5)$$

Saklı katman için aktivasyon değerlerinin ifadesi aynı şekilde elde edilebilir.

$$O_j = \frac{1}{1 + e^{-net_j}} \quad (6)$$

$$net_j = \sum_i W_{ij} O_i \quad (7)$$

Ağırlıkların yenilenmesi

$$W_{jk} = W_{jk} + \Delta W_{jk} \quad (8)$$

eşitliği ile gerçekleştirilir. Burada  $\Delta W_{jk}$  ağırlık yenileme değeridir. Geri yayılım algoritmasında ortalama kare hatası olarak bilinen hata kriteri kullanılabilir.

$$E = \frac{1}{2} \sum_p \sum_k (t_{pk} - o_{pk})^2 \quad (9)$$

Hataların karesi alınarak beklenen değerden uzak olan çıkış değerlerinin toplam hatayı oluşturması sağlanmaktadır. Hatayı minimum yapmak amacı ile hatanın ağırlıklara olan bağımlılığı hesaplanır ve gradiente bağlı olarak ağırlıklar hatayı düşürecek şekilde yenilir.

$$\Delta W_{jk} = -\eta (\partial E / \partial W_{jk}) \quad (10)$$

zincir kuralı kullanarak diferansiyel denklem çözümü aşağıdaki şekilde elde edilir.

$$\partial E / \partial W_{jk} = \delta_k O_j \quad (11)$$

Bu eşitlik denklem (10)' de yerine konursa ağırlık yenileme değeri ifadesi elde edilir.

$$\Delta W_{jk} = -\eta \delta_k O_j \quad (12)$$

$$\Delta W_{ij} = -\eta \delta_j O_i \quad (13)$$

Burada  $\delta_k$  ve  $\delta_j$  sırası ile çıkış ve saklı katman için hata terimi,  $\eta$  ise öğrenme oranıdır. Çıkış katmanı için hata terimi

$$\delta_k = (t_k - O_k) f'(net_k) \quad (14)$$

ve saklı katman için hata terimi ise aşağıdaki gibi hesaplanır.

$$\delta_j = f'(net_j) \sum_k \delta_k W_{kj} \quad (15)$$

Yukarıdaki ifadelerde  $f'()$  katmanlar arası sigmoid aktivasyon fonksiyonunun türevidir.

Elde edilen ifadelerde yapay sinir ağı eğitiminde data giriş setinin ağırlıkların etkisi tam olarak görülememektedir. Herbir ağırlıklı bağlantı için algoritmanın üreteceği ağırlık yenileme işareti tek tek incelenir ise bağıntıların her bir katman nöral aktivasyon seviyeleri dikkate alınarak verilmesi daha uygun olacaktır.

$$\Delta W_{jk} = -\eta O'_k \Delta O_k O_j \quad (16)$$

$$\Delta W_{ij} = -\eta O'_j \sum_k O'_k W_{jk} \Delta O_k O_j \quad (17)$$

$$\delta_k = O_k (1 - O_k) (t_k - O_k) \quad (18)$$

$$\delta_j = O_j (1 - O_j) \sum_k \delta_k W_{jk} \quad (19)$$

$$O_k = f\left(\sum_j O_j W_{jk}\right) \quad (20)$$

$$O_j = f\left(\sum_i O_i W_{ij}\right) \quad (21)$$

Burada  $\Delta O_k = (t_k - O_k)$

$f()$  : sigmoid aktivasyon fonksiyonu

$\delta$  : delta hata ifadesi

$\eta$  : öğrenme oranı

$M$  : çıkış katmanı neuron sayısı

$t_k$  : hedef değer

$O_k$  : çıkış aktivasyon seviyesi

$O'_k$  : çıkış aktivasyon seviyesinin türevi

$W_{jk}$  : saklı katman-çıkış katmanı arasında

ağırlıklı bağlantı

$\Delta W_{jk}$  : saklı katman çıkış katmanı arasında ağırlıklı bağlantılar için ağırlık yenileme işareti

$W_{ij}$  : giriş ve saklı katman arasında ağırlıklı bağlantı

$\Delta W_{ij}$  : giriş ve saklı katman arasında ağırlıklı bağlantılar için ağırlık yenileme işareti

Elde edilen son ifadelerden yapay sinir ağında giriş değerlerinin ağırlıklı bağlantıların belirlenmesinde ve dolayısı ile nöral eğitimde önemli bir rol oynadığı görülmektedir. Bu etki bir 3 bit pariti problemi için incelenecektir. Bu amaç için pariti bitte lojik 0 ve lojik 1 durumları için kullanılan semboller (-3, 3) aralığında farklı değerler olarak alınacaktır. Giriş datalarının bu şekilde düzenlenmesi sonucunda girişi oluşturan veri setinin istatistiksel değerleri; ortalama değeri ve standart sapması düzenli olarak değiştirilecektir. Bir sonraki bölümde sunulan deneysel çalışmalarda elde edilen bu data setleri için her bir katmanlarda farklı

aktivasyon fonksiyonları kullanıp bir sonraki katmana giriş olarak yönlendirilecek değerin dinamik değişim aralığını ve dolayısı ile standard sapmasını farklı kılınarak nöral öğrenmenin aktivasyon fonksiyonuna olan bağımlılığında incelenecektir.

### 3. DENEYSSEL SONUÇLAR

Yapay sinir ağı, geri yayılım algoritması kullanılarak üç girişli tek çıkışlı pariti bit problemi için eğitilmiştir. Ağ iki katmanlı olup bu katmanlar için farklı aktivasyon fonksiyonları kullanılmıştır. Uygulama Matlab Neural Network Toolbox kullanılarak gerçekleştirilmiştir. Aktivasyon fonksiyonlarına göre 4 farklı ağ konfigürasyonu Toolbox'ta mevcut olan *newff* fonksiyonu kullanılarak gerçekleştirildi. Örnek pariti bit problemi için kullanılacak semboller [-3 3] aralığında 0,2'lik adımla değiştirilerek toplam 464 adet birbirinden farklı giriş verisi elde edildi. Elde edilen 464 giriş veri seti 4 ağ konfigürasyonu için kullanılarak toplam 1856 nöral öğrenme simule edildi. Ayrıca nöral öğrenmenin başlangıç ağırlık değerlerine

olan bağımlılığını minimuma indirmek için 8 farklı başlangıç ağırlık değerleri Tablo 1' de gösterildiği gibi seçilerek toplam 14848 eğitim simülasyonu gerçekleştirildi.

4 farklı ağ konfigürasyonu için aktivasyon fonksiyonlarının kombinasyonu Tablo 1.de 2. sütünde verilmiştir. Verilen birinci fonksiyon ağı giriş ve saklı katman arası, ikinci fonksiyon ise saklı katman ile çıkış katmanı arasındaki aktivasyon fonksiyonudur. Bu durumda Tablo 2'de verilmiş olan 4 ağ konfigürasyonu oluşturulmuştur.

Tablo 2. Eğitim için oluşturulan ağ konfigürasyonları

Ağ Konfigürasyonu	I. Fonksiyon	II. Fonksiyon
I.	Tansig	Logsig
II.	Logsig	Logsig
III.	Tansig	Purelin
IV.	Logsig	Purelin

Tablo 1. Eğitilen ağlardan elde edilen bütün sonuçlar

Ağırlık Başlangıç Aralığı	Aktivasyon Fonksiyonu		Eğitilebilen Ağlar İçin	
	1 Katman	2.Katman	Eğitilebilen Ağ Sayısı	Ortalama İterasyon
$W_o$				
$W_a = -0,1$ $W_{\bar{u}} = 0,1$	Tansig	Logsig	431	559
	Logsig	Logsig	423	929,5
	Tansig	Purelin	418	615,9
	Logsig	Purelin	315	1138
$W_a = -0,25$ $W_{\bar{u}} = 0,25$	Tansig	Logsig	428	564,05
	Logsig	Logsig	434	851,97
	Tansig	Purelin	414	443,89
$W_a = -0,5$ $W_{\bar{u}} = 0,5$	Logsig	Purelin	369	824,28
	Tansig	Logsig	441	567,79
	Logsig	Logsig	440	779,18
	Tansig	Purelin	428	642,3
$W_a = -0,75$ $W_{\bar{u}} = 0,75$	Logsig	Purelin	412	489,5
	Tansig	Logsig	443	556,95
	Logsig	Logsig	429	669,18
$W_a = -1$ $W_{\bar{u}} = 1$	Tansig	Purelin	426	443,61
	Logsig	Purelin	409	577,31
	Tansig	Logsig	445	568,01
$W_a = -1,25$ $W_{\bar{u}} = 1,25$	Logsig	Logsig	435	672,1
	Tansig	Purelin	423	439,83
	Logsig	Purelin	415	567,15
	Tansig	Logsig	446	589,9
$W_a = -1,5$ $W_{\bar{u}} = 1,5$	Logsig	Logsig	433	607,13
	Tansig	Purelin	417	417,52
	Logsig	Purelin	416	513,85
	Tansig	Logsig	439	556,6
$W_a = -1,75$ $W_{\bar{u}} = 1,75$	Logsig	Logsig	436	626,33
	Tansig	Purelin	416	442,65
	Logsig	Purelin	418	499,69
	Tansig	Logsig	433	546,85
$W_a = -1,75$ $W_{\bar{u}} = 1,75$	Logsig	Logsig	441	655,07
	Tansig	Purelin	406	501,45
	Logsig	Purelin	416	468,2

Eğitim sonrasında elde edilen sonuçlar aktivasyon fonksiyonları farklı olan ağların verilen giriş aralıklarından kaçında eğitildiği, her bir eğitilen ağ yapısına göre iterasyon sayılarının ortalama değeri ve farklı giriş aralıklarının ağ eğitimine olan etkisini gözlemleme olanağı verecek şekilde Tablo 1’de verilmiştir

Tablo 1’in birinci sütünü ağların başlangıç ağırlıklarının değişim aralıklarını, iki ve üçüncü sütün her bir ağı katmanları arasında seçilen aktivasyon fonksiyonlarının konfigürasyonunu, dördüncü sütün her bir ağı 464 giriş veri seti için eğitilebilen ağ sayısını, beşinci sütün eğitilebilen ağlar için ortalama iterasyon sayılarının değerini vermektedir. Aktivasyon fonksiyonunun nöral öğrenme üzerindeki etkisini daha açık görebilmek için Tablo 1’de son iki sütünde verilen değerlerin ortalaması Tablo 3’de verilmiştir. Burada açık olarak I ve II konfigürasyonuna sahip ağların için eğitilebilen ağ sayısı diğer ağlardan daha fazladır. Ancak ortalama iterasyon sayısı göz önüne alındığında II konfigürasyona sahip ağlarının performansı çok düşük gözükmektedir. Öğretilebilen ağ sayısı ve ortalama iterasyon kriterleri göz önüne alındığında en kötü sonuçlar Tablo 3’ e göre IV. Konfigürasyona sahip ağlar için ortaya çıkmaktadır.

Tablo 3 Her bir ağ konfigürasyonu için ortalama eğitilebilen ağ sayısı ve iterasyon

YSA	I	II	III	IV
Ortalama giriş seti sayısı	438,25	433	418,5	396
Ortalama iterasyon sayısı	563	723	493	573

Buna karşılık Tansig-Logsig ve Tansig-Purelin aktivasyon fonksiyonlarının kombinasyonunun kullanıldığı I. ve III. ağ yapıları için nöral eğitimin diğerlerine nazaran yüksek performans gösterdiği gözlemlenmiştir.

Yapay sinir ağı eğitiminin giriş veri setine olan bağımlılığını incelemek için ise için Tablo 4. verilen 6 farklı aralığa sahip girişler için eğitilebilen ağların ortalama iterasyon sayısı verilmiştir

Tablo 4. Giriş data setine bağlı olarak eğitilebilen her bir ağ için ortalama iterasyon sayıları

Giriş sembolleri	I	II	III	IV
[-2.4 2.4]	338	427	85	141
[-2 2.],	351	436	60	184
[-1.4 1.4]	355	434	67	191
[-1 1]	368	435	76	158
[-0.4 0.4]	414	476	119	454

Tablodan giriş veri setinin nöral öğrenme üzerindeki etkisi açık olarak tüm ağ yapıları için görülmektedir. Bütün ağ yapıları için giriş veri setinde semboller için kullanılan değerler arasındaki fark azaldıkça ağ eğitimi zorlaşmakta iterasyon sayısı her bir ağ için artmaktadır. Burada verilen sonuçlardan da aynı giriş veri setleri en hızlı eğitimin III. ağ konfigürasyonu için elde edilmekte olduğu görülmektedir.

#### 4. SONUÇ

Bu çalışmada bir çok katmanlı YSA için katmanlar arası aktivasyon fonksiyonlarının ve istatistiksel özellikleri değiştirilmiş aynı probleme ait giriş veri setinin yapay sinir ağı eğitimine etkisi araştırılmıştır. Bir 3-bit pariti problemi için kullanılan sembollerin {a<sub>i</sub>} temsil ettiği değerler data setinin ortalama değeri ve standart sapma değeri düzenli olarak değiştirilerek elde edilen 464 farklı giriş seti, 8 farklı başlangıç noktası ve 4 farklı ağ konfigürasyonu oluşturarak toplam 14848 farklı eğitim denemesi gerçekleştirilmiş ve geri yayılım algoritmasının performansı her bir durum için ölçülmüştür. Sonuçlar III. konfigürasyon olarak adlandırılan ve Tansig-Purelin aktivasyon fonksiyonlarının kombinasyonunu kullanan ağ yapısının ortalama iterasyon sayısı ve öğretilebilen ağ sayısı açısından en iyi performansı gösterdiğini ortaya koymuştur. Yapılan deneysel çalışmalar ayrıca giriş veri setinin istatistiksel değerlerinin eğitim üzerindeki etkisini de açık olarak ortaya koymaktadır.

#### KAYNAKLAR

- [1] Kadiyalam S., and Gurumorthy R., Neural Network with modified backpropagation learning applied to structural optimisation, AIAA JOURNAL, 34, pp. 408-412, 1996
- [2] Jean J.S.N. and Wang J., Weight smoothing to improve network generalisation, IEEE TRANSACTIONS ON NEURAL NETWORKS, 5 (5), pp.752-763 1994
- [3] Reed R. Pruning algorithms- A survey, IEEE TRANS. ON NEURAL NETWORKS Vol. 4 (2), pp. 740-747, 1996
- [4] H.C. Chua, Solving Two-Spiral Problem Through Input Data Encoding, ELECTRONIC LETTERS, Vol. 31(10), pp. 813-814, 1995
- [5] H. Altun and K. M. Curtis, Exploiting the Statistical Characteristics of the Speech Signals for improvement Neural Learning in MLP Neural Networks, IEEE WORKSHOP ON NEURAL NETWORKS FOR SIGNAL PROCESSING, NNSP'98, pp. 547-556, Cambridge, UK, August 1998
- [6] Han J., Moraga C., The Influence of the Sigmoid function Parameters on the Speed of Backpropagation Learning, LECTURE NOTES IN COMPUTER SCIENCE, Vol. 930, pp. 195-201, Dothmund Germany