

# SENSÖRLERİN DOĞRUSAL OLMAYAN EĞRİLERİNİN YAPAY SİNİR AĞLARIYLA ELDE EDİLMESİ

Turgut KIRCI  
TSE Metroloji ve Kalibrasyon  
Daire Başkanlığı  
Elektronik Mühendisi

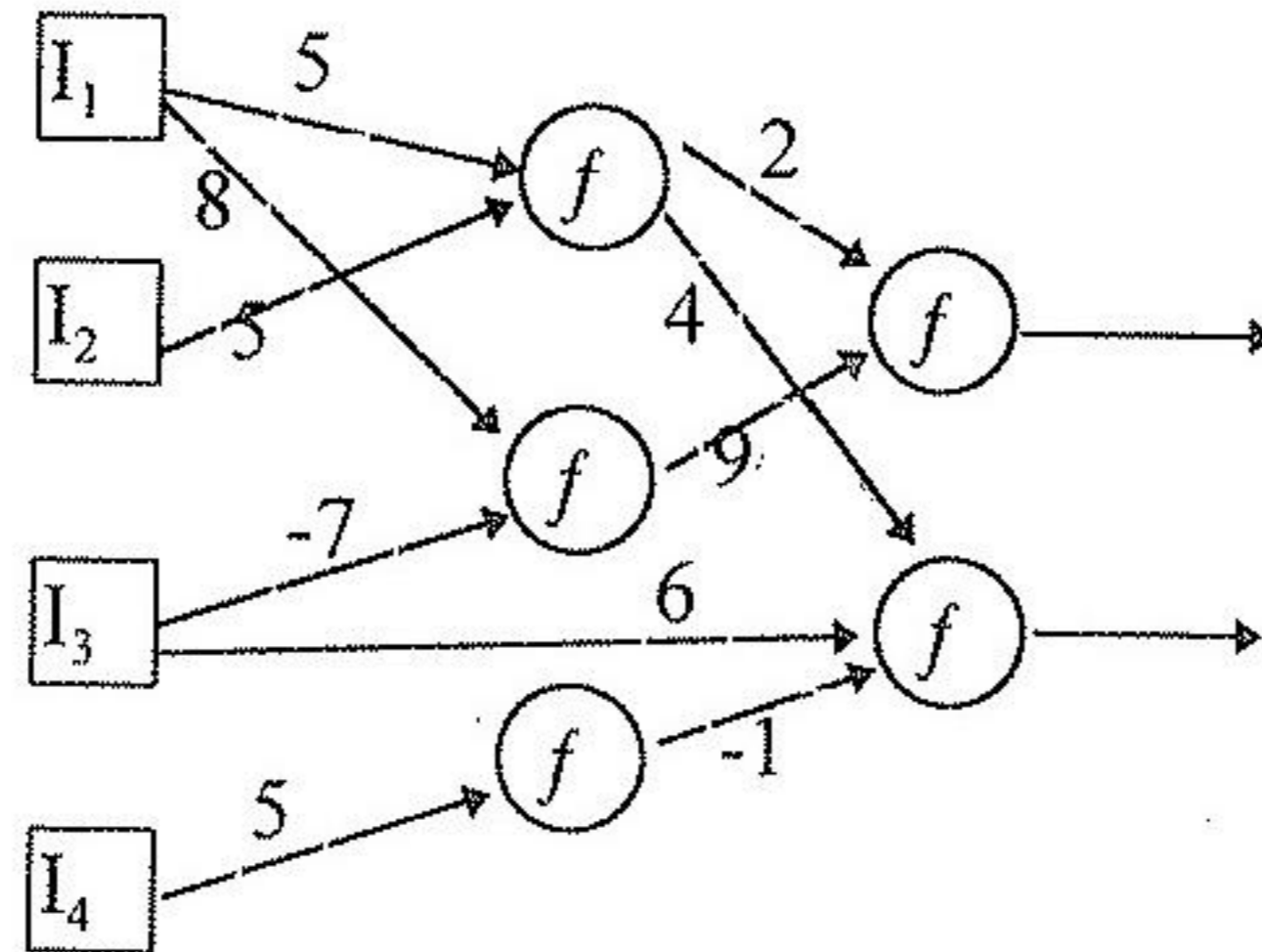
İbrahim ÖZTÜRK  
Başkent Üniversitesi  
Öğretim Görevlisi

## 1. GİRİŞ

Ölçüm teknolojisinde kullanılan sensörlerin çoğu doğrusal olmayan giriş-çıkış karakteristiklerine sahiptir. Ölçümler, bu sensörlerin çıkışındaki genellikle elektriksel olan büyüklüğü, ideal sensör tipine göre değerlendiren gösterge sistemleriyle mümkün olmaktadır. Mevcut sensör eğrilerinin ideal sensör eğrilerinden sapmış olması, bu sensörler ideal sensör tipine göre üretilmiş gösterge sistemleriyle birlikte kullanıldığında yanlış ölçümler yapılmasına sebep olmaktadır. Kalibrasyon neticesinde ortaya konabilen bu sapma hassas ölçümlerin yapıldığı ve kalibrasyon raporunun her zaman değerlendirilmesinin mümkün olmadığı ölçümlerde problem kaynağı olmaktadır. Yapay sinir ağlarıyla sisteme, sensörün kendine özgü, ideal tipten farklı, sensörün kalibrasyonu sırasında ortaya konmuş giriş-çıkış eğrisi öğretilmekte, kalibrasyon raporundan bağımsız doğru ölçümler mümkün olabilmektedir.

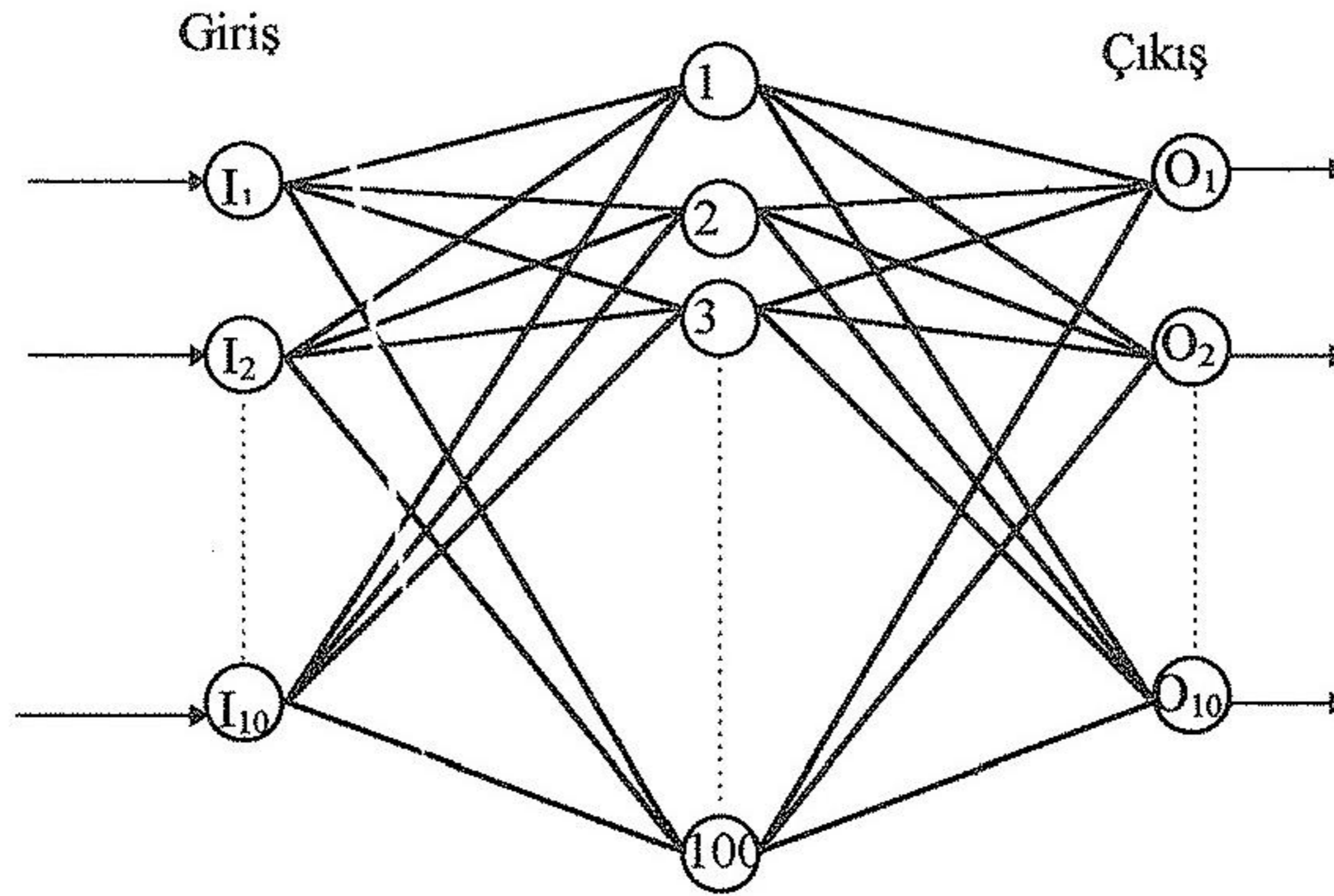
## 2. YAPAY SİNİR AĞLARI VE EĞRİNİN ELDE EDİLMESİ

Yapay sinir ağları canlıların sinir sisteminden ilham alınarak ortaya konmuş bir modeldir. Basitçe; bir giriş kümesi (input) girişin değerlendirildiği hücreler ve hücreleri kendi arasında ve girişe bağlayan, giriş değerini bir katsayı ile çarpıp ileten yapay ağlardan oluşur. Şekil 1.



Şekil 1 Yapay Sinir Ağı

Biz örnek çalışmamızı K tipi ısı çift üzerinde yaptık. Giriş kümesi olarak 0 -1000°C arası 10°C'nin katlarını aldık. Çoklu bir giriş kümesi oluşturabilmek için bu değerleri ikili tabana çevirip kullandık. Ağımızı iki katman (layer) ve sonuçları yine ikili tabanda verecek şekilde tasarladık. Girişte sıcaklık değerlerini 10 basamakla ifade ettik. Çıkışta da 10 basamak kullandık. Bu şekilde çıkış çözünürlüğü onluk tabanda 3 basamağa düştü. Örneğin 500°C için 20100µV'u, 1000°C için 41300µV'u karşılık aldık. (Son iki basamak her zaman sıfırlardan oluştu.) Ara katmanı 100 hücre kullanarak oluşturduk.(Şekil 2) Ağa eğriyi öğretmek içinse backpropagation algoritmasını<sup>(1,2)</sup> kullandık.



Şekil 2

### Backpropagation Algoritması<sup>(1,2)</sup>:

- 1- Giriş vektörü ağa uygulanır.  $x_p(x_{p1}, \dots, x_{pN})$
- 2- Ara katmana uygulanacak vektör hesaplanır

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi}$$

- 3- Ara katmanın çıkışları hesaplanır.

$$i_{pj} = f_j^h(net_{pj}^h)$$

- 4- Çıkış bulunur.

$$net_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj}$$

- 5-Net çıkış değeri çıkış tanımlanan fonksiyondan geçirelerek bulunur.

$$o_{pk} = f_k^o(net_{pk}^o)$$

- 6- Hata bulunur.

$$\delta_{pk}^o = (y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o)$$

7- Ara katman için hata bulunur.

$$\delta_{pj}^h = f_j^{h'}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o$$

8- Öğrenmeyi sağlayan katsayılar hatayı azalatacak yönde yenilenir.

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i$$

9- Bu işleme çıkıştaki hata belli bir seviyenin altına ininceye kadar devam edilir.

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2$$

Hücrelerde girişin geçirildiği fonksiyon sigmoid'tir.

$$f(x) = \frac{1}{1+e^x}$$

### 3. SONUÇ

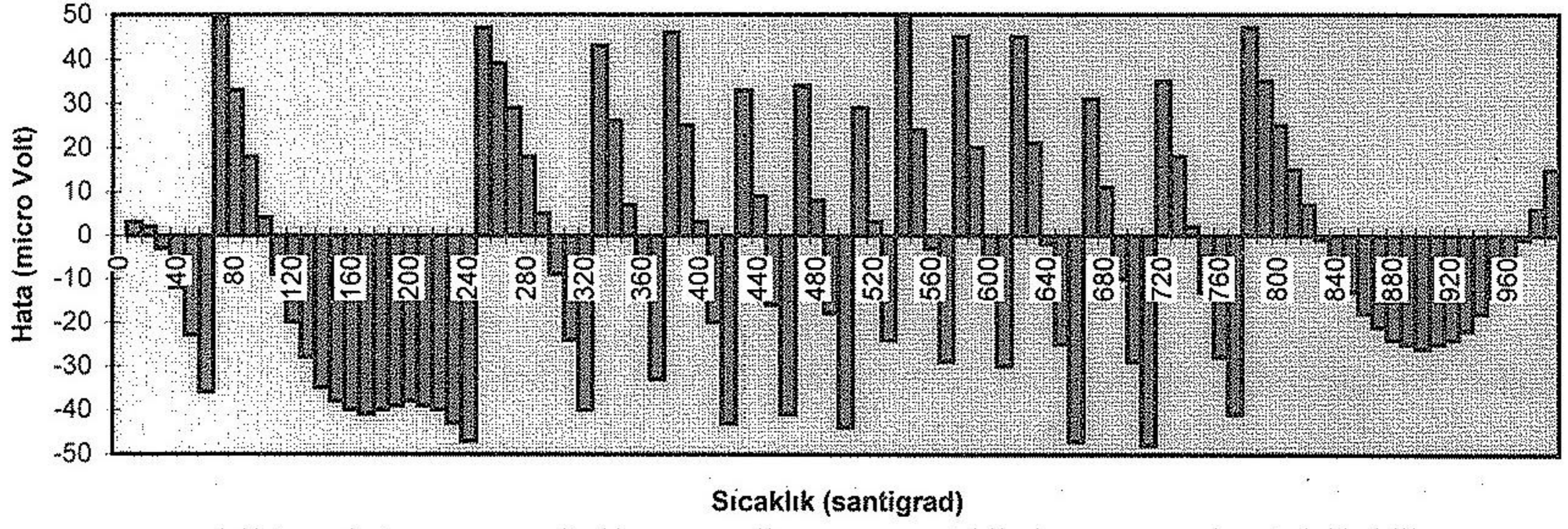
Ağ eğriyi 3 basamak çözünürlükte %100 başarıyla öğrendi. Biz yine de elde ettiğimiz eğriyi 5 basamak çözünürlükte verilmiş eğriyle karşılaştırarak hatasını bulduk. (Grafik) Gerekli durumlarda ağdaki katman sayısı ve/veya hücre sayısı artırılarak daha iyi çözünürlükte de aynı başarı sağlanabilir. Fakat bu durumda da öğrenme süresinin artması kaçınılmazdır.

Burda uyguladığımız bu metod genel olarak sensörlerin bir üst standartla karşılaştırılıp bulunan hatalarının azaltılması ve daha hassas ölçümler yapılmasında elverişli gözükmektedir. Herhangi bir şekilde sensör tipine bağlılık da söz konusu değildir. Sistemin dezavantajı ise ölçüm sistemini kompleksleştirip ek maliyet getirmesidir.

### KAYNAKLAR

- 1-Neural Network Algorithms, Applications and Programming Techniques. J. A. Freeman, D.M. Skatoro. Addison Wesley Publishing Company 1991.
- 2- An Introduction to Computing with Neural Network. R. T. Lippmann. IEEE ASSP Magazine April 1987 Sayfa:4-22.

Hata Grafiği



Hata Oranı

