

***BIL684***  
***Nöron Ağları***  
***Dönem Projesi***

***SNNS Uygulama Parametrelerinin bir Örnek Aracılığı ile  
İncelenmesi***

Bu rapor ile Bil684 Nöron Ağları dersi kapsamında gerçekleştirilmiş olan "SNNS Uygulama Parametrelerinin bir Örnek Aracılığı ile İncelenmesi" projesinin sonuçlarını dökümanete edecektir. Proje bağlamında ağ topolojisinin, nöron sayılarının, öğrenme fonksiyonlarının öğrenme ve hata yakalama üzerindeki etkileri incelenecektir.

### Genel Tanım

Tanımlamalar esnasında örnek olarak seçilen örnek karakter tanımlama olarak belirlenmiştir. 5\*7'lik bir matris olarak tanımlanan karakter matris'i yalnızca 1 ve 0'lerden oluşmaktadır. Tasarlanan sistem'de ara bir değer söz konusu değildir. Her girdinin yalnızca 0 ve 1 olması söz konusudur.

Bu amaç ile hazırlanan örnek girdi kütüğü Şekil 1'de bir kesiti verilmiş olan kütüktür.

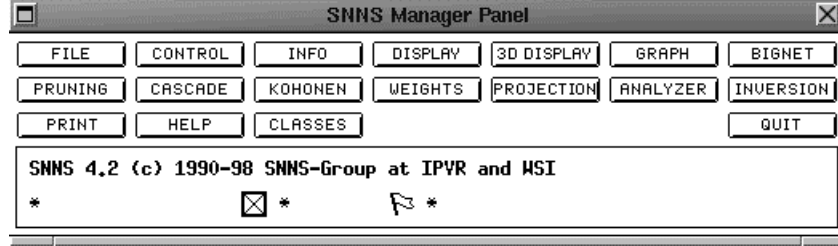
```
SNNS pattern definition file V3.2
generated at Mon Apr 25 18:08:50 1994

No. of patterns : 36
No. of input units : 35
No. of output units : 8
# Input pattern 1:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 0 0 0 1
1 0 0 0 1
# Output pattern 1
0 1 0 0 0 0 0 1
# Input pattern 2:
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
1 0 0 0 1
1 0 0 0 1
1 1 1 1 0
# Output pattern 2
0 1 0 0 0 0 1 0
# Input pattern 3:
0 1 1 1 0
1 0 0 0 1
1 0 0 0 0
1 0 0 0 0
1 0 0 0 0
1 0 0 0 1
0 1 1 1 0
# Output pattern 3
0 1 0 0 0 0 1 1
```

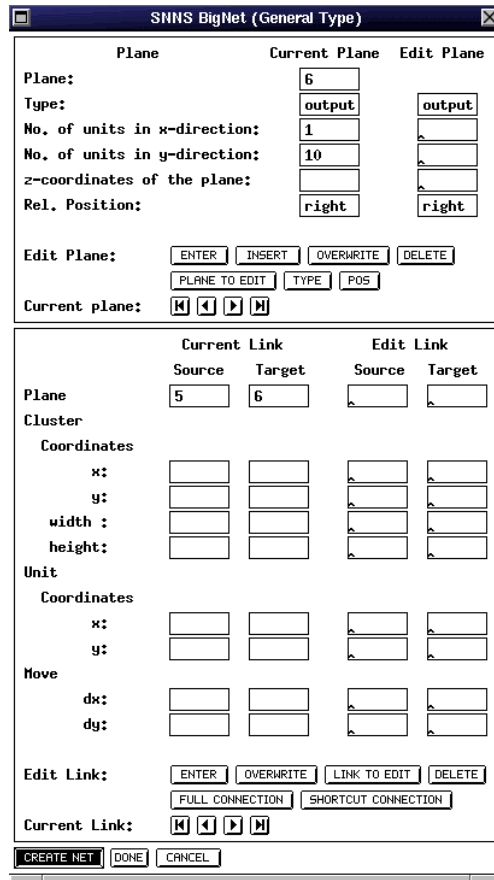
Şekil 1: Örnek Giriş Kütüğü

## BIL684 Nöron Ağları – Proje Raporu

Giriş kütüğümüzü hazırladıktan sonra SNNS çalıştırılarak ağ yaratılmıştır. Bu işlem için SNNS ilk çalıştığında görülen Şekil 2'deki "BigNet" tuşundan general seçilmelidir. "BigNet" seçildikten sonra ekrana gelen ekran'da (Şekil 3) sabit girdimiz ve çıktımız ile ilgili nöronların sayıları girilir ki bu konumuz olan sistem için 5\*7 girdi, 1\*8 çıktı matrisimizdir. Tüm ağlarımız öntanımlı olarak tam bağlantılı (full connection) olacaktır.



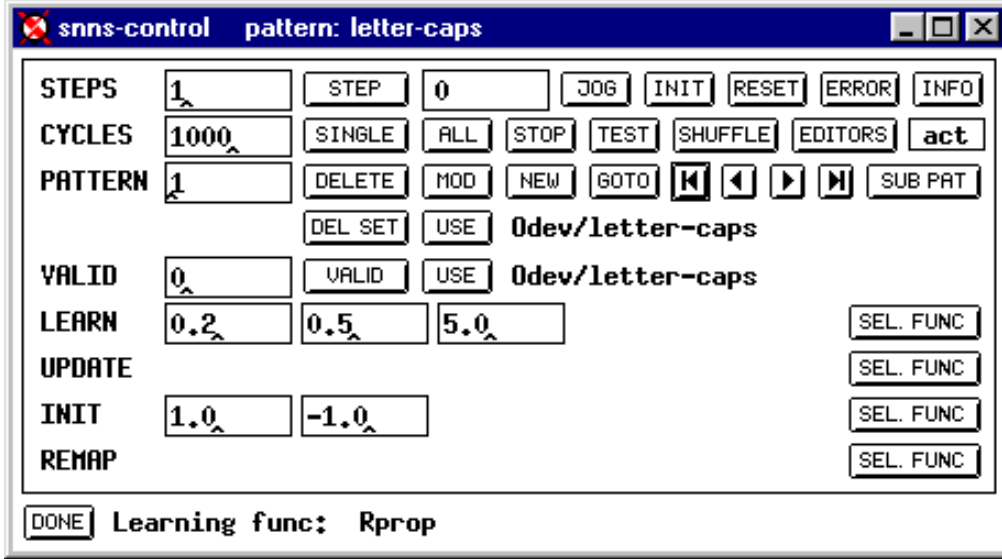
Şekil 2: SNNS Ana Kontrol Penceresi



Şekil 3: Ağ Yaratma Penceresi

### Nöron Sayısı Üzerine Denemeler;

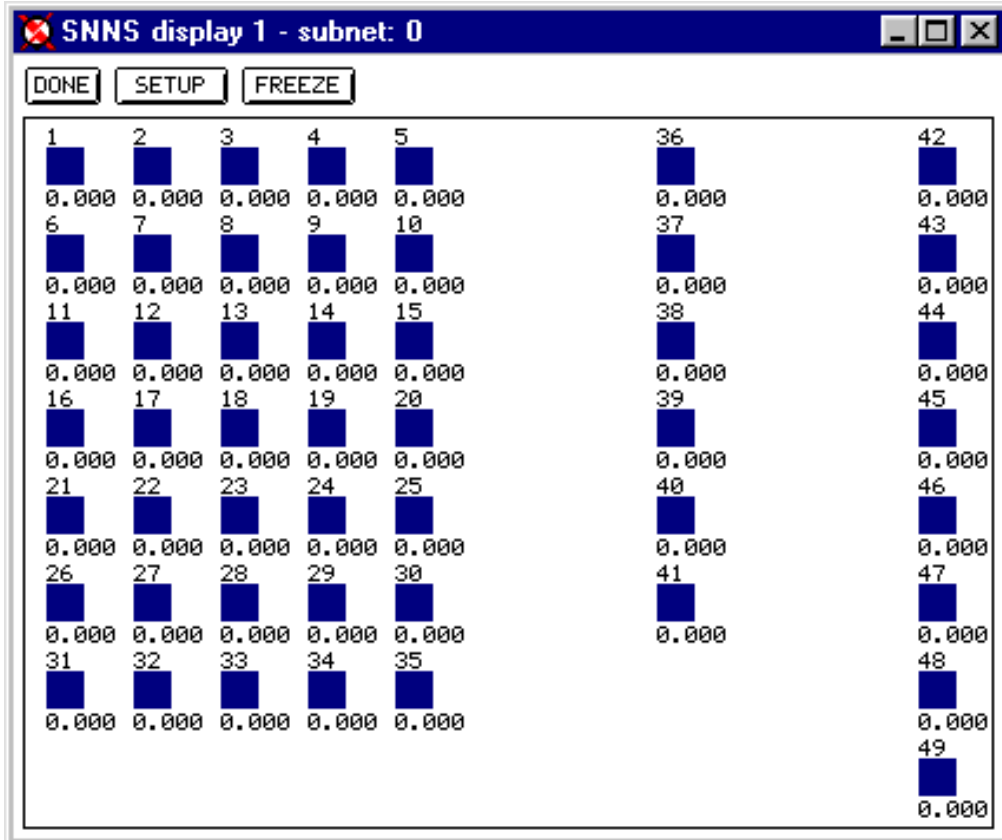
Nöron Sayıları üzerinde yapacağımız değişikliklerde etkili bir sonuç alabilmek için öğrenme fonksiyonun sabit tutulması gerekmektedir. Bu nedenle nöron sayıları üzerinde yapılan testlerde öğrenme fonksiyonu ve parametreleri sabitlenmiş olup, öğrenme fonksiyonu olarak "Rprop", parametreleri ise "0,2", "0,5", "5,0" olarak seçilmiştir. Bahsi geçen fonksiyonlar ve değerleri Şekil 4'de gösterilen ve Şekil 2'deki ana kontrol penceresindeki "control" tuşu ile açılan pencere aracılığı ile yapılmaktadır.



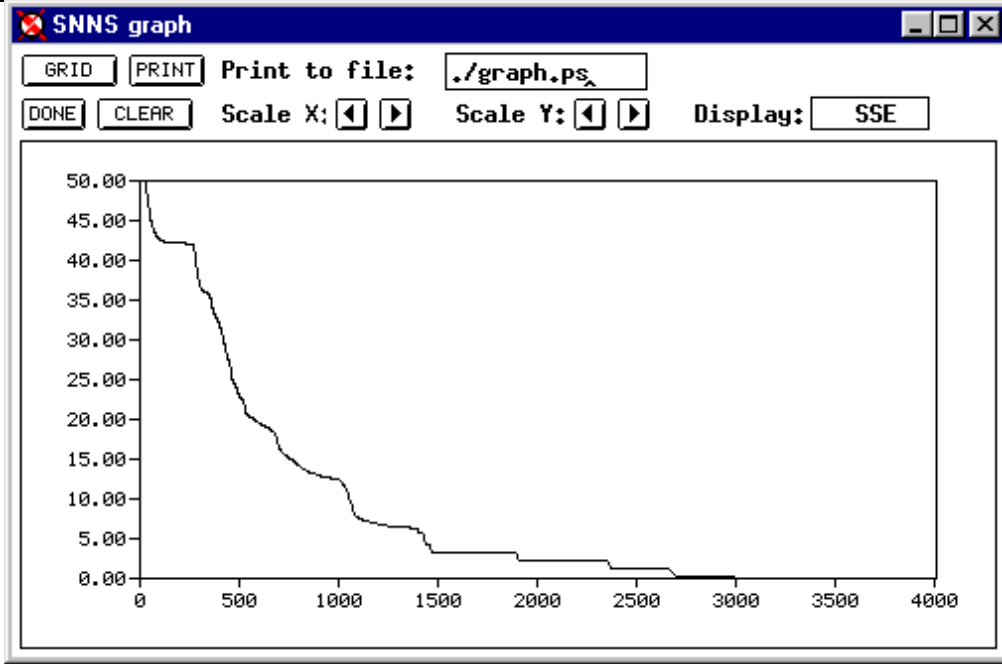
Şekil 4: Öğrenme Süreci Kontrol Penceresi

## A) 6 (Altı) Nöron

Gizli katman'da (hidden layer) 6 nöron ile yapılan denemelerde (Şekil 5'deki gösterim) öğrenmenin başarılı olabilmesi için 2700 devir (cycle) gerekli olduğu saptanmıştır (Şekil 6). Bu 2700 devirin sonucunda sistem kendisine verilmiş olan 26 harfi ve 10 rakamı %100 başarı ile tanıyabilmekte ve %6 oranındaki rastgele bit değişiminde kendini koruyabilmekte, dolayısı ile %6 oranında gürültü düzeltmesi yapabilmektedir. Fakat bu sistem yarım-harf gibi harfin yalnızca bir kısmının görüldüğü durumlarda başarısız olmaktadır.



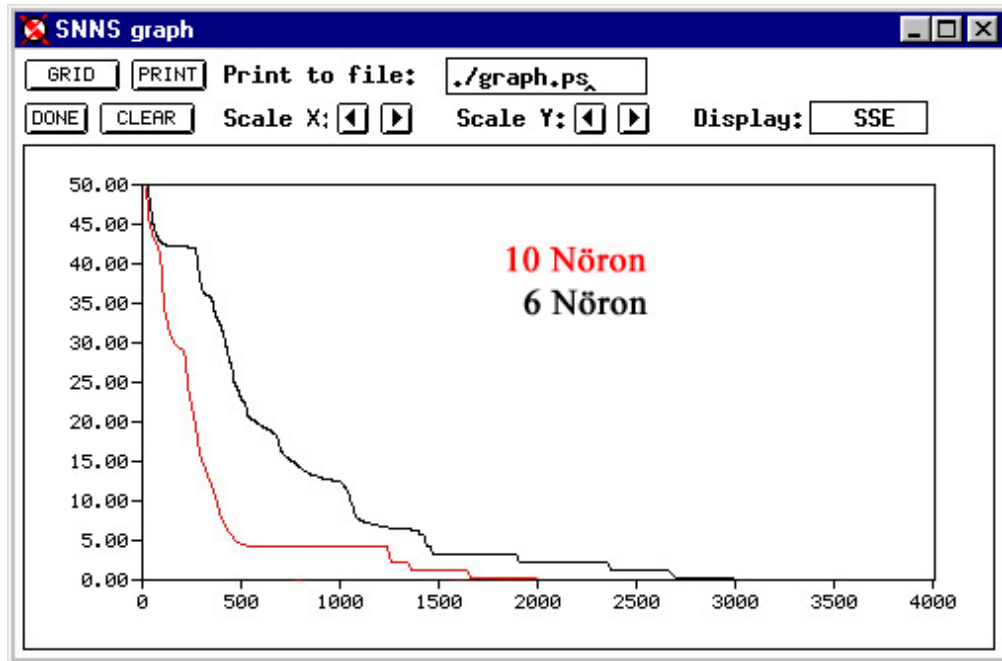
Şekil 5: 6 Nöron Kullanılarak Oluşturulan Ağ



Şekil 6: 6 Nöron İle Öğrenme Süreci Grafiği

## B ) 10 (On) Nöron

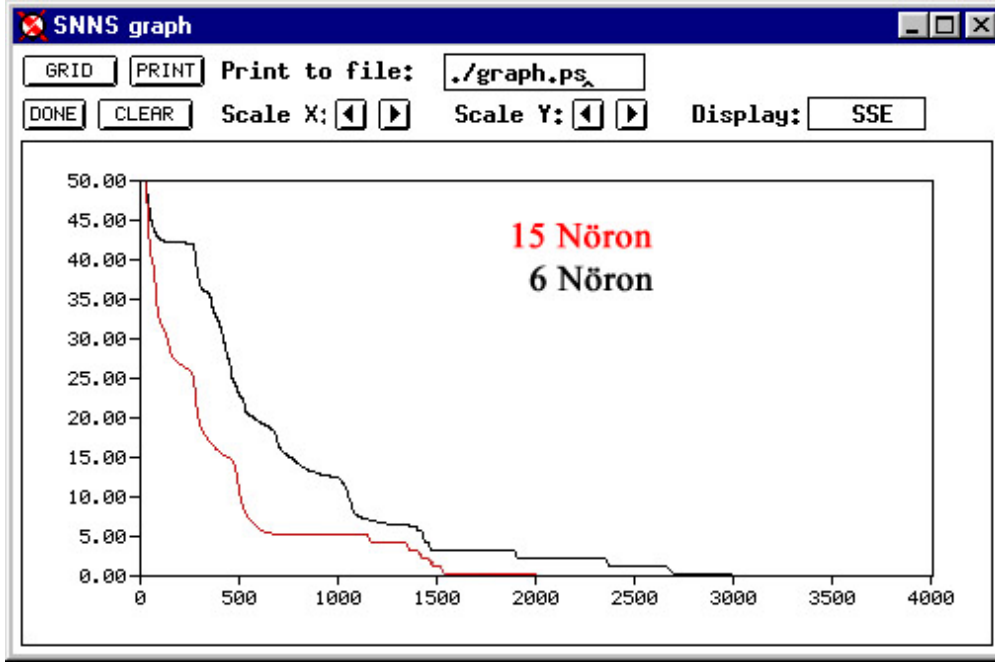
Gizli katman'daki nöron sayısının 6'dan 10'a yükseltilmesi sonucunda öğrenmenin başarılı olabilmesi için 1750 devir gerekli olduğu saptanmıştır. 6 Nöron ile 10 Nöron arasındaki öğrenme farkını Şekil 7'deki grafik aracılığı ile gösterilmektedir. Şekilde kırmızı çizgi 10 nöronlu ağın öğrenme grafiğini, siyah çizgi ise 6 nöronlu ağın öğrenme grafiğini göstermektedir. Öğrenme süresince gerçekleşen 1750 devirin sonucunda sistem kendisine verilmiş olan 26 harfi ve 10 rakamı %100 başarı ile tanıyabilmekte ve %15 oranında gürültü düzeltmesi yapabilmektedir. Fakat bu sistem yarım-harf gibi harfin yalnızca bir kısmının görüldüğü durumlarda başarısız olmaktadır.



Şekil 7: 10 Nöron İle 6 Nöron Öğrenme Süreci Grafiği Karşılaştırmaları

### C ) 15 (Onbeş) Nöron

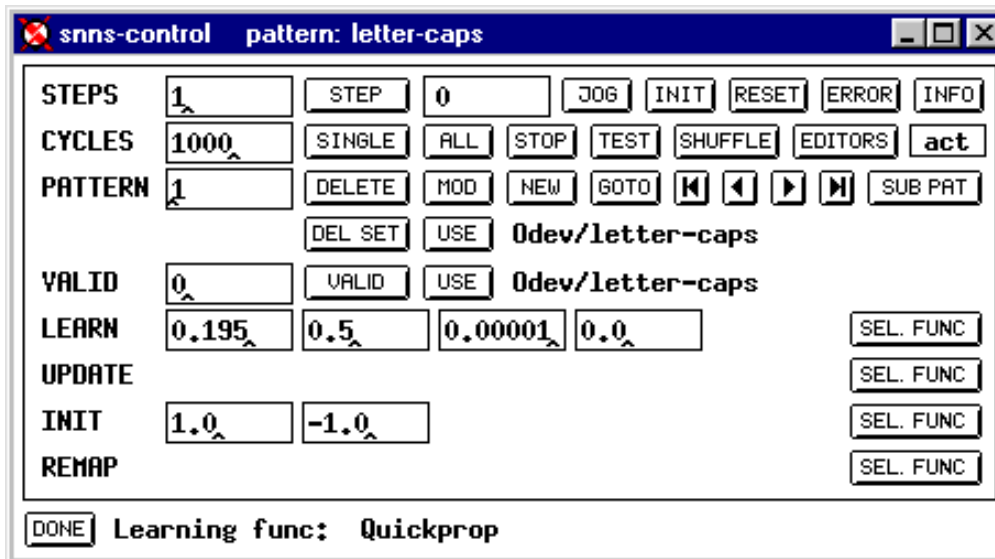
Gizli katman'daki nöron sayısının 6'dan 15'2 yükseltilmesi sonucunda öğrenmenin başarılı olabilmesi için 1510 devir gerekli olduğu saptanmıştır. 6 Nöron ile 15 Nöron arasındaki öğrenme farkını Şekil 8'deki grafik aracılığı ile gösterilmektedir. Şekilde kırmızı çizgi 15 nöronlu ağın öğrenme grafiğini, siyah çizgi ise 6 nöronlu ağın öğrenme grafiğini göstermektedir. Öğrenme sürecince gerçekleşen 1510 devirin sonucunda sistem kendisine verilmiş olan 26 harfi ve 10 rakamı %100 başarı ile tanıyabilmekte ve %21 oranında gürültü düzeltmesi yapabilmektedir. Ayrıca bu sistem yarım-harf gibi yalnızca bir kısmı görünen harflaride %83 oranında tanıyabilmektedir. Hata yaptığı harfler ise A ve B gibi birbirine çok yakın olan harflerdir.



Şekil 8: 15 Nöron İle 6 Nöron Öğrenme Süreci Grafiği Karşılaştırmaları

### Öğrenme Fonksyonu Üzerindeki Denemeler

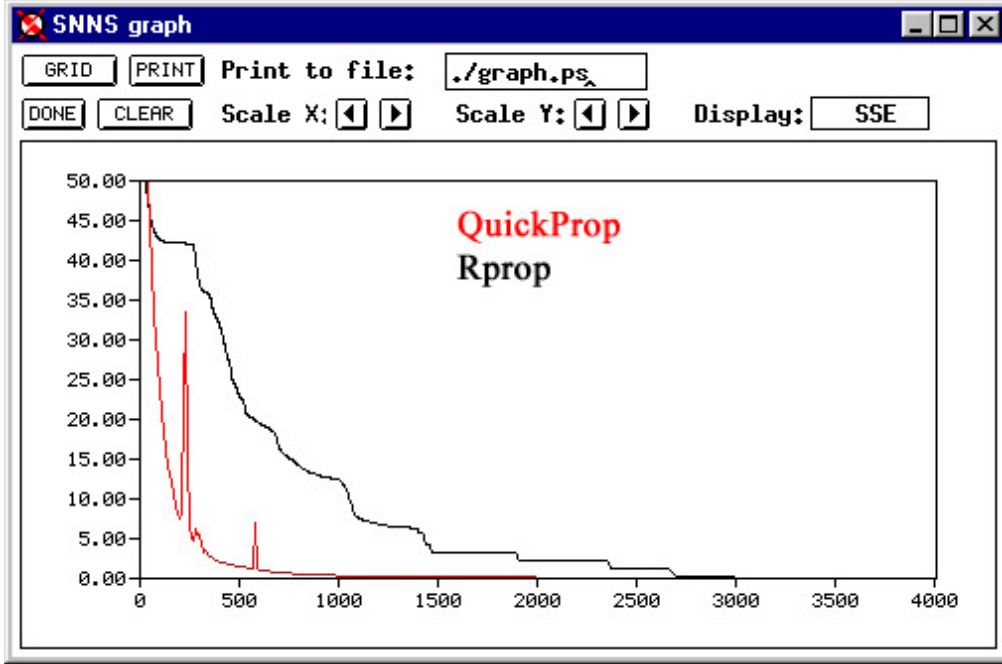
Sağlıklı ölçüm yapılabilmesi ve aradaki farkların belirlenebilmesi amacı ile ilk test koşulu olan 6 Nöron kullanılmış ve Öğrenme fonksyonu Şekil 9'da belirtildiği gibi QuickProp seçilmiş, öğrenme katsayıları ise "0,195", "0,5", "0,00001", "0,0" olarak seçilmiştir.



Şekil 9: QuickProp İçin Öğrenme Süreci Kontrol Penceresi

## A ) 6 (Altı) Nöron

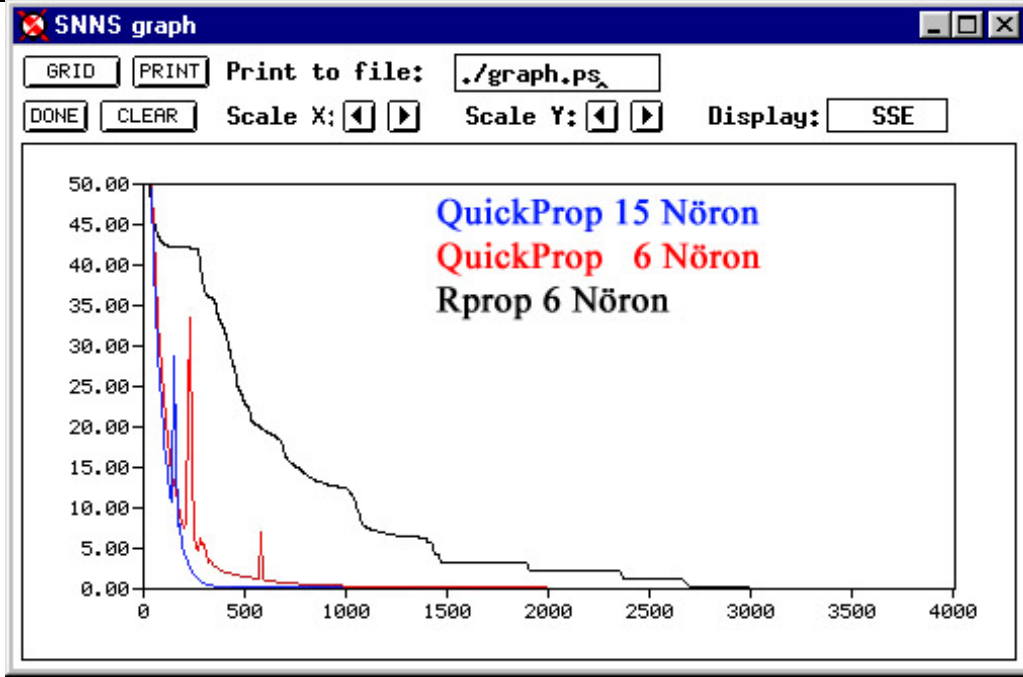
Gizli katman'da 6 nöron ve "QuickProp" öğrenme fonksyonu ile yapılan denemelerde öğrenmenin başarılı olabilmesi için 990 devir gerekli olduğu saptanmıştır. Rprop ve QuickProp öğrenme fonksyonları grafiklerinin karşılaştırılması Şekil 10'da detaylı olarak verilmiştir. Öğrenme sürecinde meydana gelen 990 devirin sonucunda sistem kendisine verilmiş olan 26 harfi ve 10 rakamı %100 başarı ile tanıyabilmektedir. Fakat bu sistem ne yazıkki herhangi bir gürültü üzerinde düzeltme yapabilme yeteneğine sahip değildir. Bu sistem çok çabuk öğrenmekte fakat bilgideki karmaşıklıklara karşı son derece zaafdır.



Şekil 10: QuickProp İle Rprop Öğrenme Süreçleri Karşılaştırmaları

## B ) 15 (Onbeş) Nöron

Gizli katman'da 15 nöron ve "QuickProp" öğrenme fonksyonu ile yapılan denemelerde öğrenmenin daha önce yapılan testler ışığında daha çabuk gerçekleşmesi beklenmektedir. Nitekim yapılan denemeler ve testler sonucunda bu doğrulanmıştır. Yapılan testlerde öğrenmenin başarılı olabilmesi için 450 devir gerekli olduğu saptanmıştır. Rprop ve QuickProp öğrenme fonksyonları grafiklerinin karşılaştırılması Şekil 11'de detaylı olarak verilmiştir. Öğrenme sürecinde meydana gelen 450 devirin sonucunda sistem kendisine verilmiş olan 26 harfi ve 10 rakamı %100 başarı ile tanıyabilmekte ve %15 oranında gürültü düzeltme yapabilmektedir.



Şekil 11: QuickProp İle Rprop Öğrenme Süreçlerinin 6 ve 15 Nöron İçin Karşılaştırmaları

## Sonuç

Yapılan tüm testlerin sonucunda aslında bildik bir gerçek ile karşılaşmıştır. Nöron sayısı ne kadar fazla olursa hata düzeltme o kadar mümkün olmaktadır. Bir diğer söyleyiş tarzı ile Nöron sayısı arttıkça sistem daha akıllı olmaktadır.

Bir diğer çıkarılabilecek sonuç ise hızlı öğrenmenin yavaş öğrenmeye oranla daha dezavantajlı olduğudur. Sistemin yavaş öğrenmesi hata düzeltme olasılığını daha yükseltmiş ve başarı oranını artırmıştır.