Artificial Neural Network Analysis of

DNA Microarray-based Prostate

Cancer Recurrence

Serkan Kurt - 08060309

Funda Karadeniz - 08060321

Harun Kara - 08060325

Prostat kanseri belirtilerinin oluşturulmasından sonra prostat kanserinin bu belirtilerle eşleştirilmesi üzerinde çalışılmıştır.Uluslararası olarak tespit edilen ve hastalığın tanısı için kabul görmüş kriterler bu projede hastalığın tesbitinde kullanılmıştır.

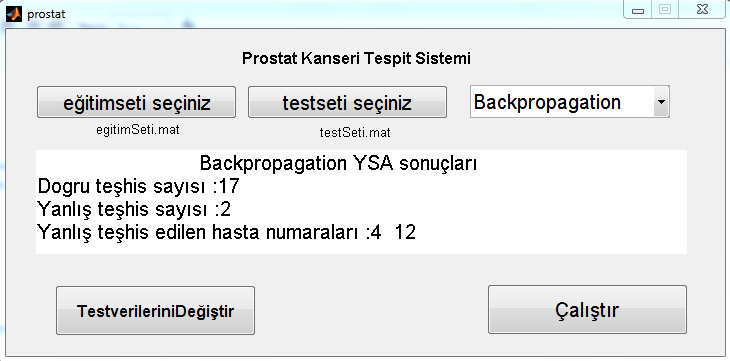
Hastalığın tanısında belirlenmiş olan 10 adet belirtinin %50 sinin görüldüğü kişilerin prostat kanseri olduğu teşhis edilmektedir.Bu kritere göre verilen hasta örneklerinin prostat kanseri olup olmadığı bilgisi ağa verilmektedir.

Bu projede geri yayılım ağları ile ağ eğitimi yapılmaktadır.Fonksiyon olarak ‘traingda’ fonksiyonu kullanılmıştır.Çıkış katmanında tek nöron kullanılmış ve hastalığı taşıyan veri seti için 1 ,hasta olmayan veri seti için 0 çıkışları vermesi istenmiştir.

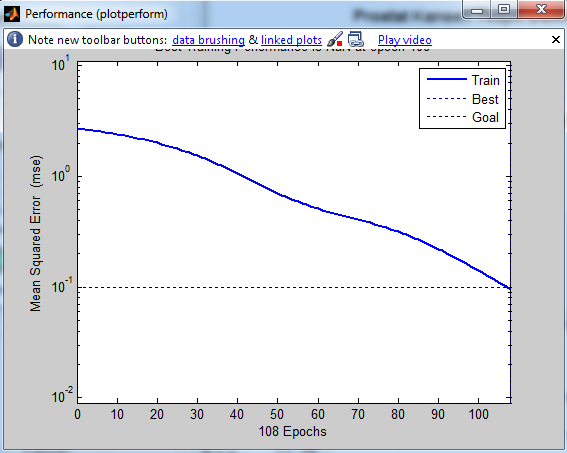
Gizli katmandaki nöron sayısının belirlenmesi için deneme-yanılma yoluyla yapılan çalışmada en iyi sonucu veren nöron sayısının 10 olduğu sonucuna varılmıştır.

45 adet eğitim verisi ve 19 adet test verisi ile yapılan testler sonucunda ortalama olarak 110 epok sonunda ortalama hata çok küçük bir değere inmekte ve istenilen değere ulaşılarak sistem eğitilmektedir.Eğitilen ağ test örneklerinin teşhisinde 1000 epok ve 10 arakatman değerleri girilğinde 19 adet test verisinin ortalama 17 tanesini doğru bilerek %90 başarı sağlamaktadır.

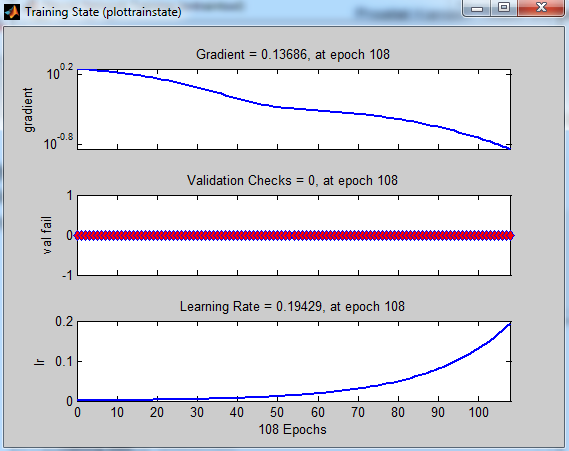
‘Backprop.m’ dosyasında projenin kodları yer almaktadır. Kolay ve anlaşılır kullanım için ‘prostat.m’ dosyasında arayüz tasarlanmıştır. ‘prostat.m’ çalıştırıldıktan sonra ‘set’ dosyasının içindeki eğitim ve test verileri girilerek yöntem olarak backpropagation seçilerek programın çalışması test edilebilir.



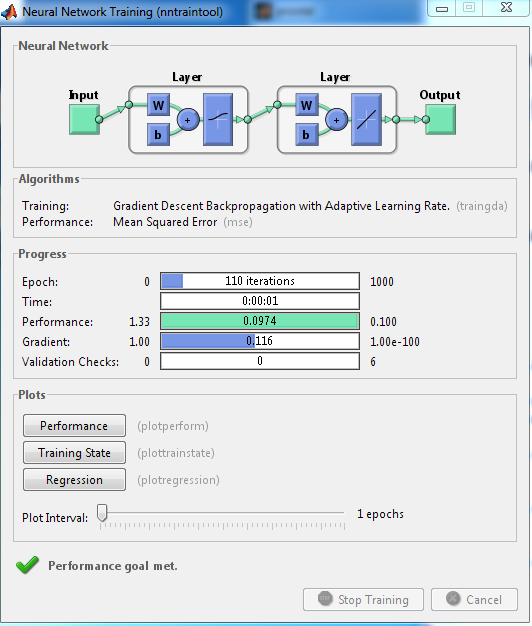
‘randtest.m’ fonksiyonu ile rasgele veri girişleri sağlanmıştır.Verileri rasgele atmak için arayüzde ‘TestverileriniDeğiştir’ butonu eklenmiştir.Bu buton ile veriler random olarak değiştirilip proğramın çalışması test edilebilir.



Ortalama karesel hata grafiğidir.Hata değeri 1 den daha büyük bir değerden başlamış ve giderek azalmıştır.110 epok sonunda programda belirlemiş olduğumuz 0.1 değerine ulaşmış ve ağın eğitimi gerçekleştirilerek program sonlandırılmıştır.



Learning Rate(Öğrenme oranı) sıfıra çok yakın bir değerden başlayarak giderek artmış ve ağ öğrenmeye başlamıştır. Validation Check grafiği programın çalışması sırasındaki programsal hatayı göstermektedir. Program sonlanana kadar hiç bir hatayla karşılaşmamıştır.



110 iterasyon sonucunda istenilen hata değerine ulaşmış ağın nntraintool grafiği.

MATLAB PROGRAM KODLARI

function [dogru, yanlis, yanlislar]=backprop(gizli, cikis, epok, p, t, ptest, ttest)

fonk='traingda';

net= newff(minmax(p), [gizli,cikis], {'logsig','purelin'},fonk);

net.trainParam.epochs =epok; %maksimum epok sayısının belirlenmesi

net.trainParam.show =250; %kaç epokta bir hata grafiği çizilecek

net.trainParam.goal = 0.1; %hedeflenen hata değeri

net.trainParam.lr = 0.001; %öğrenme oranı

net = init(net);

net = train(net,p,t);%eğitim için ayrılan verilerle sistemin eğitilmesi

sonuc = sim(net,ptest);%eğitimin ardından,ağın eğitim verilerine yanıtı

ncol = size(sonuc);

%test verilerinin ne kadarına doğru ne kadarına yanlış yanıt

%verildiğinin tesbiti

fark = round(sonuc) - ttest;

dogru = 0;

yanlis = 0;

ndx = 0;

for i=1:ncol(2)

if fark(i) == 0

dogru =dogru + 1;

else

yanlis =yanlis +1;

ndx =ndx +1;

yanlislar(ndx) = i;

end

end

end