# *FANNTool 1.0*

## Kullanım Kılavuzu

**Giriş** :

Yapay Sinir Ağları ilhamını beyin hücresi nöronların çalışma prensibinden almış, ve yaygın kullanım alanına sahip bir Yapay Zeka tekniğidir.

FANN ( Fast Artificial Neural Network Library ) Yapay Sinir Ağlarının en yaygın olarak kullanılan Çok katmanlı İleri beslemeli Geri yayılmı ( Multi-Layer Feed Forward Backpropagate) türünün kullanımı için C dilinde yazılmış bir Kütüphanedir. Cross-platformdur. Ve Perl, PHP, Java, Delphi ve C# gibi pekçok programlama dili ile de kullanılabilir. Lisansı LGPL dir ki Ticari uygulamalarınızda bile kullanabilirsiniz. Ve pek çok YSA ktüphanesinden çok daha hızlıdır.

FANNTool ise tarafımızdan yazılan FANN kütüphesi imkanlarını Görsel olarak kullamanızı sağlayan bir programdır. Bu program ile ;
Verilerinizi FANN kütüphanesi formatında hazırlanmasını

- YSA dizaynını
- Eğitimini
- Testini
- Çalıştıramasını

yapabilirsiniz.

En Son FannTool 1.0 yayınlandı. Fakat tanıtımı ve kullanımının dökümantasyonu konusunda şikayetler var. Kullanımının öğretmek için bu dökümanı hazıladık. İnşallah bir faydası dokunur

Kuru kuruya menüler ve seçenekler üzerine açıklama yazmak yerine uygulama yaparak anlatmayı tercih ediyoruz. örneğimize başlayalım.

**Amaç** : Elimizde 1980 den 2006 yılına kadar olan aylık ortalama güneş lekesi sayısı var. geçmiş aylara bakarak sonraki ayın değerini tahmmin eden bir Yapay sinir ağı dizayn etmeye çalışacağız. Güneş Lekeleri nedir ne önemi var gibi sorularınız için wikipediadan bir alıntı yapalım

*Uygun filtrelemeyle Güneş gözlemlendiğinde ilk dikkati çeken etrafına göre daha soğuk olması nedeniyle daha koyu görüken belirli sınırlara sahip güneş lekeleridir. Güneş lekeleri, güçlü manyetik kuvvetlerin ısıyayımı engellediği ve sıcak iç bölgeden yüzeye doğru enerji transferinin azaldığı yoğun manyetik etkinliğin olduğu bölgelerdir. Manyetik alan koronanın aşırı ısınmasına neden olur ve yoğun güneş püskürtüleri ile koronada kütle fırlatılmasına neden olan etkin bölgeler oluşturur.*

*Güneş'in üzerinde görünür güneş lekelerinin sayısı sabit değildir ama Güneş döngüsü denen 11 yıllık bir döngü içinde değişiklik gösterir. Döngünün tipik minimum döneminde çok az güneş lekesi görünür ve hatta bazen hiç görünmez. Gözükenler yüksek enlemlerde bulunur. Güneş döngüsü ilerledikçe Spörer yasasının açıkladığı gibi güneş lekelerinin sayısı artar ve ekvatora doğru yaklaşır. Güneş lekeleri genelde zıt manyetik kutuplara sahip çiftler olarak bulunur. Ana güneş lekesinin manyetik polaritesi her güneş döngüsünde değişir, dolayısıyla bir döngüde kuzey manyetik kutba sahip olan leke bir sonraki döngüde güney manyetik kutba sahip olur.*

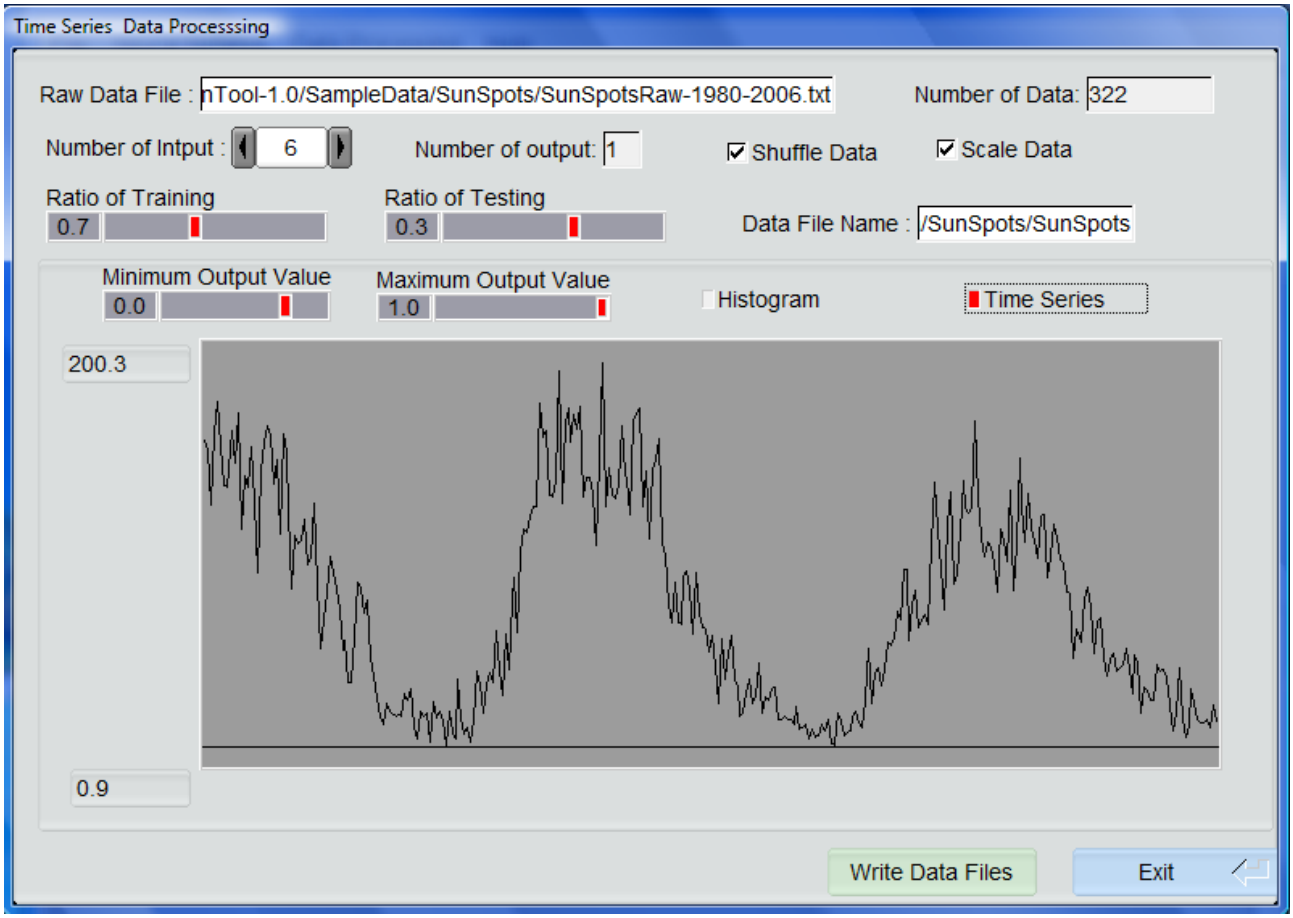
*Son 250 yılda gözlemlenen güneş lekelerinin tarihi, ~11 yıllık güneş döngüsü görülebilmektedir.*

*Güneş döngüsünün uzayın durumu üzerinde büyük etkisi vardır, ve Dünya'nın iklimi üzerinde de önemli bir etki yapar. Güneş etkinliğinin minimumda olduğu dönemler soğuk hava sıcaklıklarıyla, normalden daha uzun süren güneş döngüleri de daha sıcak hava sıcaklıklarıyla ilişkilendirilir. 17. yüzyılda güneş döngüsünün bir kaç on yıl boyunca tamamen durduğu gözlemlenmiştir; bu dönemde çok az güneş lekesi görülmüştür. Küçük Buz Çağı ya da Maunder minimumu diye bilinen bu dönemde Avrupa'da çok soğuk hava sıcaklıklarıyla karşılaşılmıştır.[35] Daha da önceleri benzer minimum dönemler ağaç halkalarının analiziyle ortaya konmuştur ve bu dönemler normalden daha düşük global hava sıcaklıklarıyla eşleşmektedir*

Elimizdeki "*SunSpotsRaw-1980-2006.txt*" dosyasında 1980 den 2006 yılına kadar olan ve her satırda 1 veri olmak üzre aylık ortalama Güneş lekesi sayısı var.

t1
t2
t3
t4
....
tn

Programımızın *Data Processing* kısmını açıp veri dosyamızı seçiyoruz. Ham verilerimizin Zaman serisi türünden olduğundan "*Time Series Data Processing*" penceresi açılır.




"*Data Processing*" in aldığı verilere ham veriler diyoruz. Çünkü

- Ölçeklendirilmiş değil :Yapay Sinir Ağları için kullanılacak veriler ( 0 , 1) veya ( -1 , 1 ) aralığına ölçeklendirilmelidir.
- Veriler FANN kütüphanesi ile doğrudan kullanılacak formatta değil. O şekilde yazdırmak lazım

*Veri Dosyasının yapısı : Veri Dosyası dediğimiz şey Giriş ve çıkış değerlerinin bulunduğu basit bir text dosyasıdır.*
*Örnek Sayısı(N) Giriş(K) Çıkış (J)*
*G11 G12 G13 ...G1K*
*Ç11 Ç12 Ç13...Ç1J*
*G21 G22 G23 ...G2K*

*Ç21 Ç22 Ç23...Ç2J*
*....*
*GN1 GN2 GN3 ...GNK*
*ÇN1 ÇN2 ÇN3...ÇNJ*

*Mesela 3 Giriş 1 Çıkışlık 2 Örnekli bir veri dosyası*

*2 3 1*
*1 1 1*
*0*
*1 0 1*
*1*

- Yapay Sinir ağları için kullanılacak verilerin Eğitim ve Test için olmak üzre en az ikiye bölünmesi lazımdır.

FannTool'un "*Data Processing*" kısmı işte bu işlemleri yapıyor.

Pek tabii ki bizim ayarlamamız gereken seçenekler var.

**Number of Input** : Giriş veri sayısını buradan seçiyoruz. Yani örneğimizde. Geçmiş kaç aylık veri ile gelecek ayın sonucunu tahmin etmek istiyoruz. Bu seçimin belli bir kuralı yok. Mesela 6 dediğimizde geçmiş 6 ay verisi giriş 7. ay çıkış diye yazılarak veri dosyası oluşturulur. Biz örenğimizde 12 kullandık.

**Minimum Output Value** : Yukarda ölçeklemeden bahsetmiştik Ölçekleme (Scale) işlemi sonrasında çıkan en küçük değer
**Maximum Output Value** : Ölçekleme (Scale) işlemi sonrasında çıkan en büyük değer

*Eğer ölçekleme aralığını ( 0 , 1 ) seçmişsek veri kümemizdeki en küçük değer 0 en büyük değer 1 seçilir. Ve hesaplama ona göre yapılır.*

**Ratio of Training** : Verinin Eğitim için ayrılacak kısmının oranı
**Ratio of Testing** : Verinin Test için ayrılacak kısmının oranı
*mesela Verilerimizin % 70 ni Eğitim için kullanmak istiyorsak Ratio of Training 0.7 olmalı*

**Schuffle** : Bu kısmı işaretlerseniz Veriler Dosyaya yazdırılmadan önce karıştırılır. Yoksa sırasıyla yazılır.

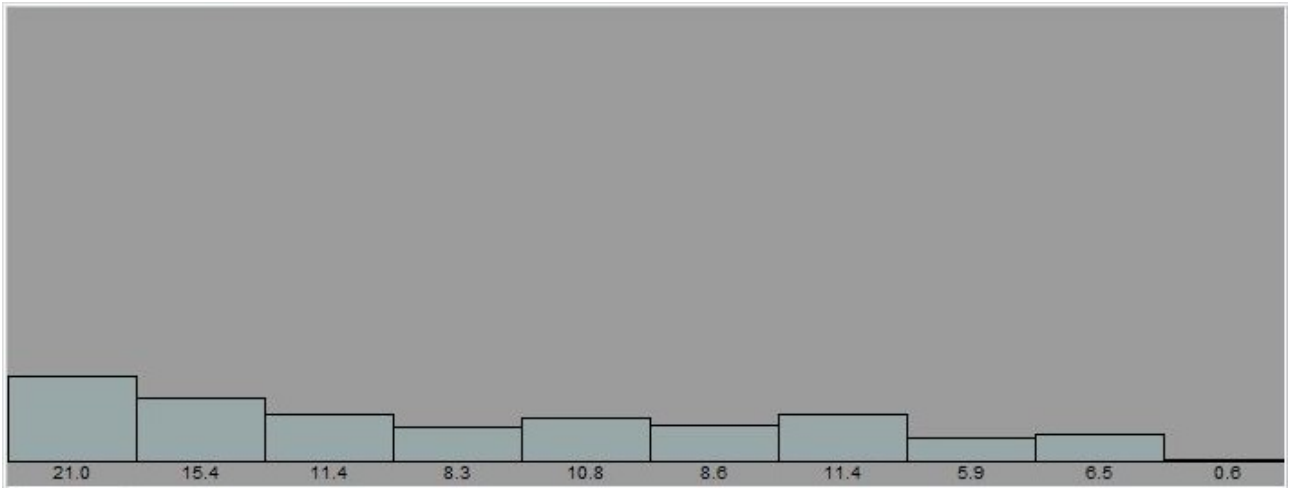
**Scale** : Bu kısmı işaretlerseniz Veriler Ölçeklendirilir. Yoksa olduğu gibi yazılır. Eğer verileriniz zaten istenen aralıkta ise Ölçeklendirmeye ihtiyacınız olmayacağından bu seçeneği işaretlemeyebilirsiniz.
Bu işlem nasıl yapılır. Değişkenin en küçük (min) ve en büyük (max) değerleri tespit edilir ve bir aralık hesaplanır
aralık = ( max-min ) // Bizim örneğimizde 200,3 – 0,9
Örneğimizde ( 0 - 1) aralığını kullandığımızdan ölçeklendirme formülümüz şöyle oluyor.

*Xö= (x-min)/( max-min )*
YSA'yı kullanırken öleçeklendirilimiş değerden gerçek değere çevirmek de gerekebilir.
Bizim örneğimiz için olanı verelim (0 1 ) aralığını kullandığımızdan

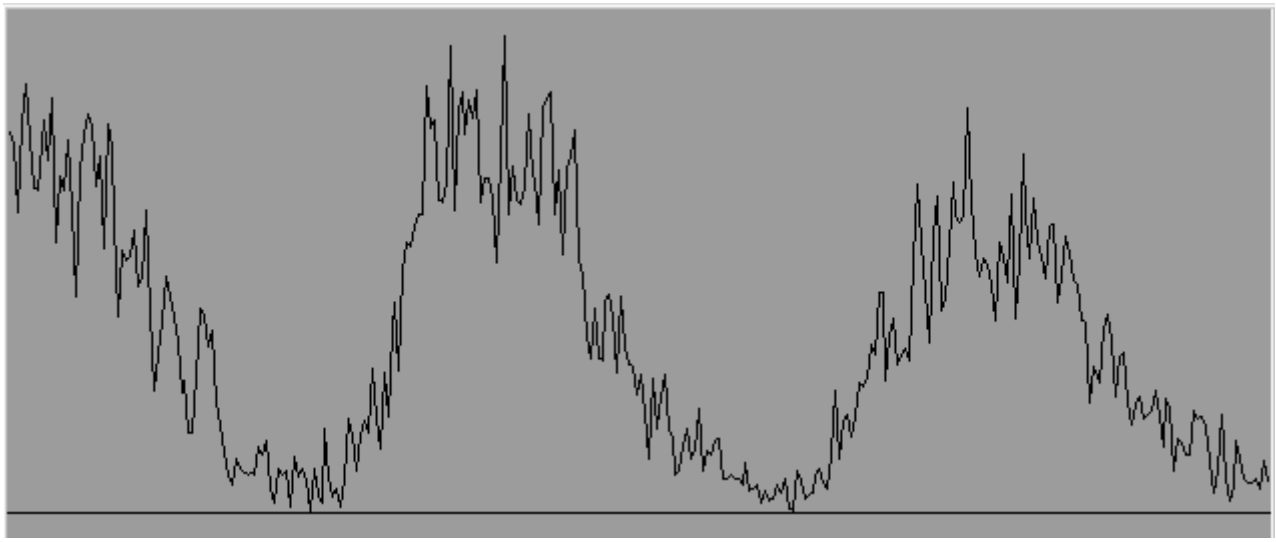
x= Xö * ( max-min ) + min

"*Time series Data Processing*" Penceresinde verilerin iki çeşit grafiğini görebilirsiniz

**Histogram**




**Zaman Serisi**



bu grafiklerin amaçları ne çeşit bir veriyle uğraştığımızı görebilmekdir. Mesela: belkide verilerimizin içinde tamamiyle yanlış bir değer var bu histogram da görülebilir ve Ham verilerimizdeki bu değereleri eleyebiliriz. ( uzun belkide başka bir yazı konusu )

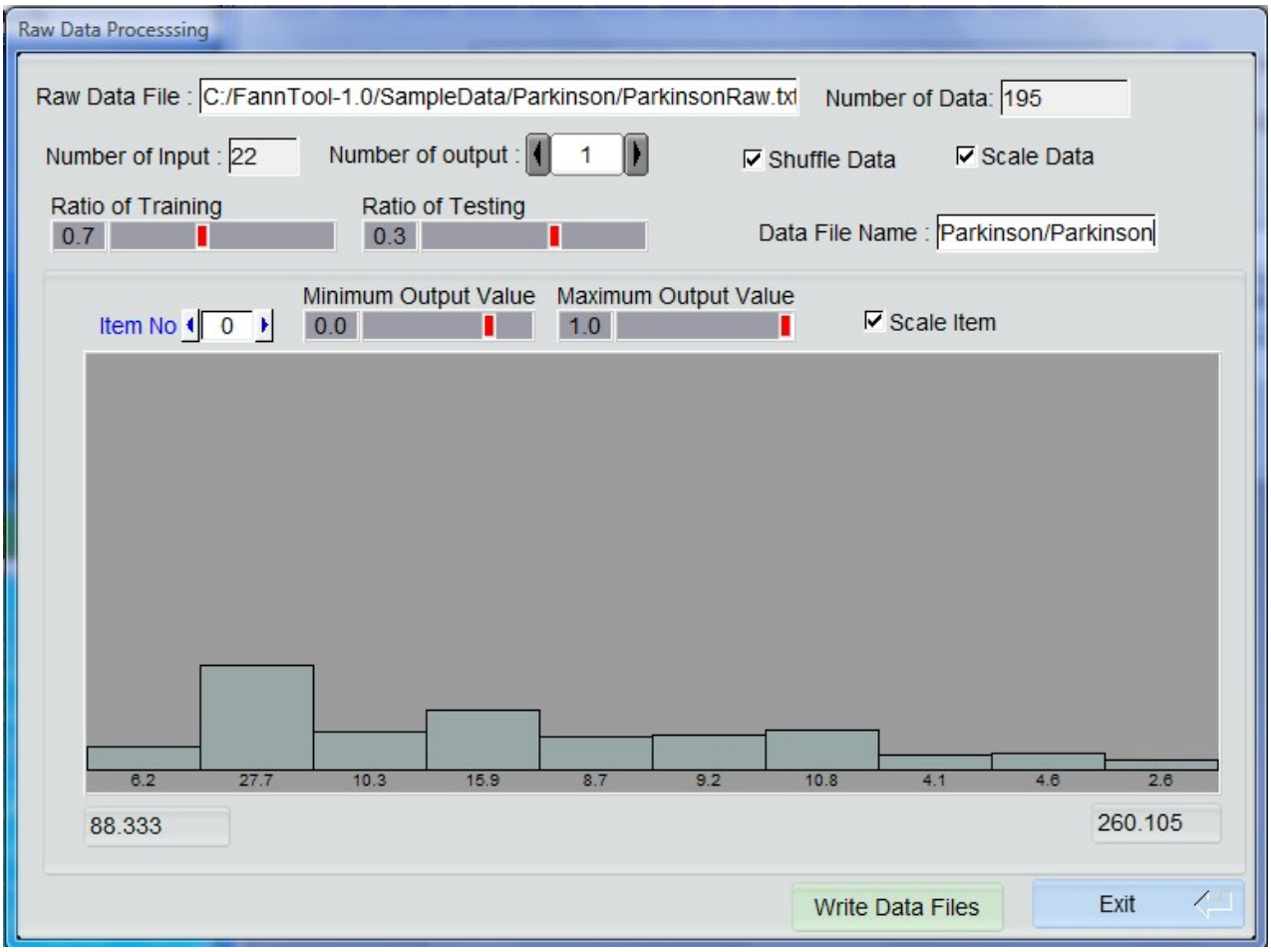
**Data File Name** : Çıkışta üretilecek olan dosyaların isimleri ve konumlarını belirliyorsunuz kendinden tanımlı "Out" için

*"Out-train.dat" : Eğitim veri Dosyası Fann formatında*
*"Out-test.dat" : Test veri Dosyası Fann formatında*
*"Out-scale.txt" : Ölçekleme için kullanılan Parametrelerin yazılıdığı metin dosyası*

dosyaları oluşturulur.

Çoğu zamana veriler zaman serisi şeklinde değildir. Matris şeklindedir. Verinizin durumuna göre Data Processing kısmı bu durumda "*Raw Data Processing*" peceresini açar

**Matris Şeklindeki Ham Veri Dataları için :**



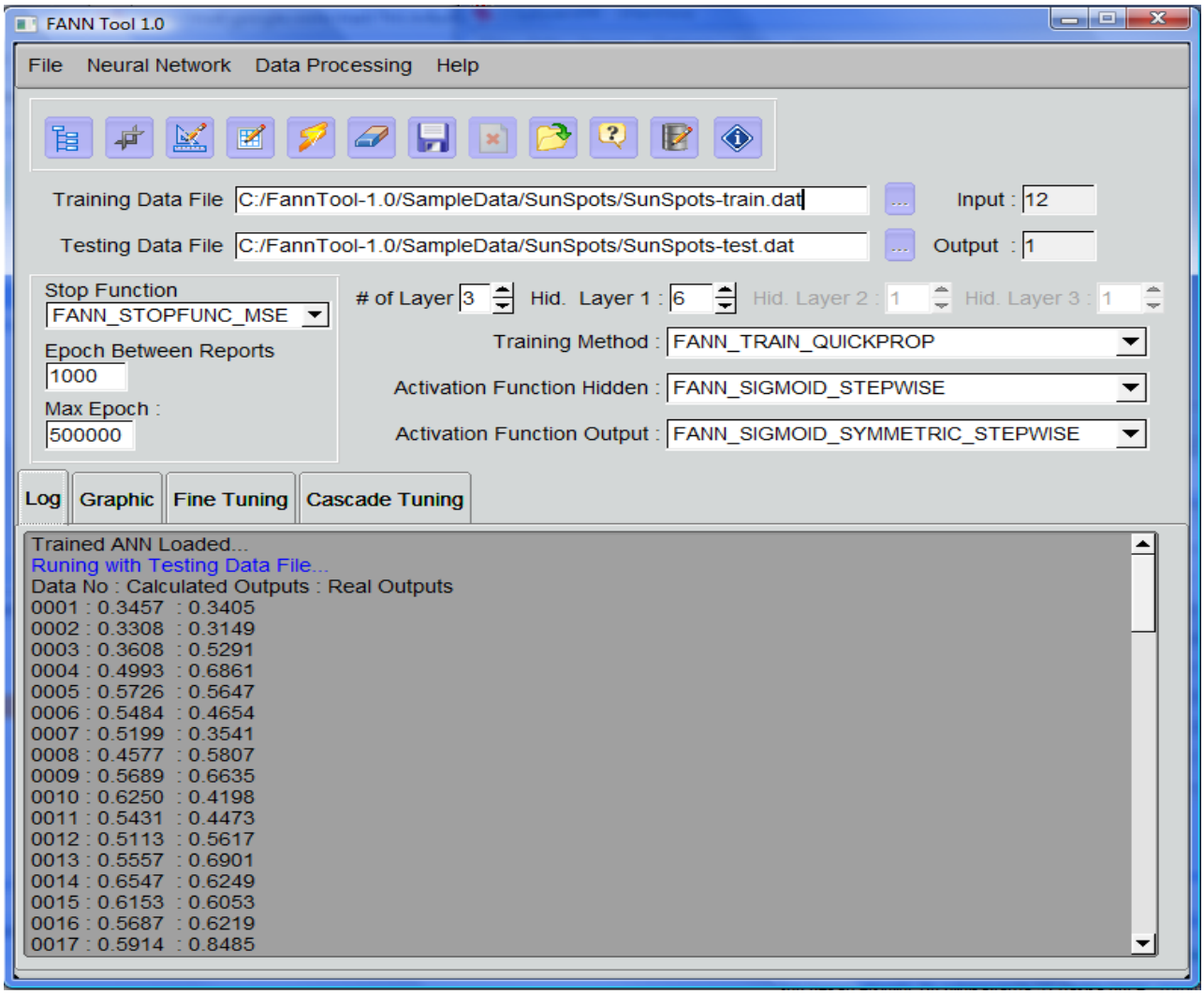

Eğer verilerimiz zaman serisi şeklinde değilse ve bir Tablolama programıyla hazırlanmış matis şeklindeyse.
Mesela X ( 3 ) giriş ve Y ( 2 ) çıkış değerleri olsun.
Dosyamızda veriler bu şekilde tutulur.
X11 X21 X31 Y11 Y21
X12 X22 X32 Y12 Y22
...
X1n X2n X3n Y1n Y2n
gibi görünür. Mesela FannTool ile birlikte yayınladığımız "ParkinsonRaw.txt" dosyası böyle bir veri içerir. 23 sütündan oluşan verinin ilk 22 si giriş ve son sutun çıkıştır. Kişinin parkinson hastası olup olmadığınaın tespitidir detaylar için buraya bakabilirsiniz. "ParkinsonRaw.txt" dosyasını açtığınızda bir parça değişik bir ekranla karşılaşırsınız. "*Raw Data Processing*" peceresi

Bu ekrandaki değişliklerden kısaca bahsedelim

**Number of Output** : Zaman sersinin aksine çıkış değerini seçmeniz gerekir. Ve En sağdaki sutunlardan kaç tanesinin çıkış değeri olduğunu seçersiniz.

**Item No** : Sutunlar birbirlerinden bağımsız değişkenler olduklarından ayrı ayrı histogramlarını görüp ayrı ayrı ölçeklendirme işlemi yapmanız gerekir.

Yukardaki işlemleri yapıp veri dosyalarımızı oluşturduğumuza göre Yapay Sinir Ağımızı Dizayn etmeye başlaya biliriz.




Önce FannTool'un seçeneklerine bir bakalım

**Training Data File** : Eğtim için kullanacağımız Veri dosyası
**Testing Data File** : Test için kullanacağımız Veri dosyası

Eğitim için gereken dosayı seçince Yapay Sinir Ağının Giriş ve Çıkış değerleri belirlenmiş olur. Örneğimizde 12 Giriş ve 1 Sonuç. Fann kütüphanesi ile - şimdilik - çok katmanlı İleri Beslemeli Geri yayılımlı Yapay Sinir Ağları yapabiliyoruz. Çok katmanlı demek en az 1. Saklı katman içeriyor demektir.

**# of Layer** : Katman sayısı = Giriş + Saklı ( Minimum 1 Maksimum 3 ) + Çıkış
**Hid Layer 1** : 1. Saklı katmandaki Nöron Sayısı
**Hid Layer 2** : 2. Saklı katmandaki Nöron Sayısı ( # of Layer >= 4 )
**Hid Layer 3** : 3. Saklı katmandaki Nöron Sayısı ( # of Layer = 5 )

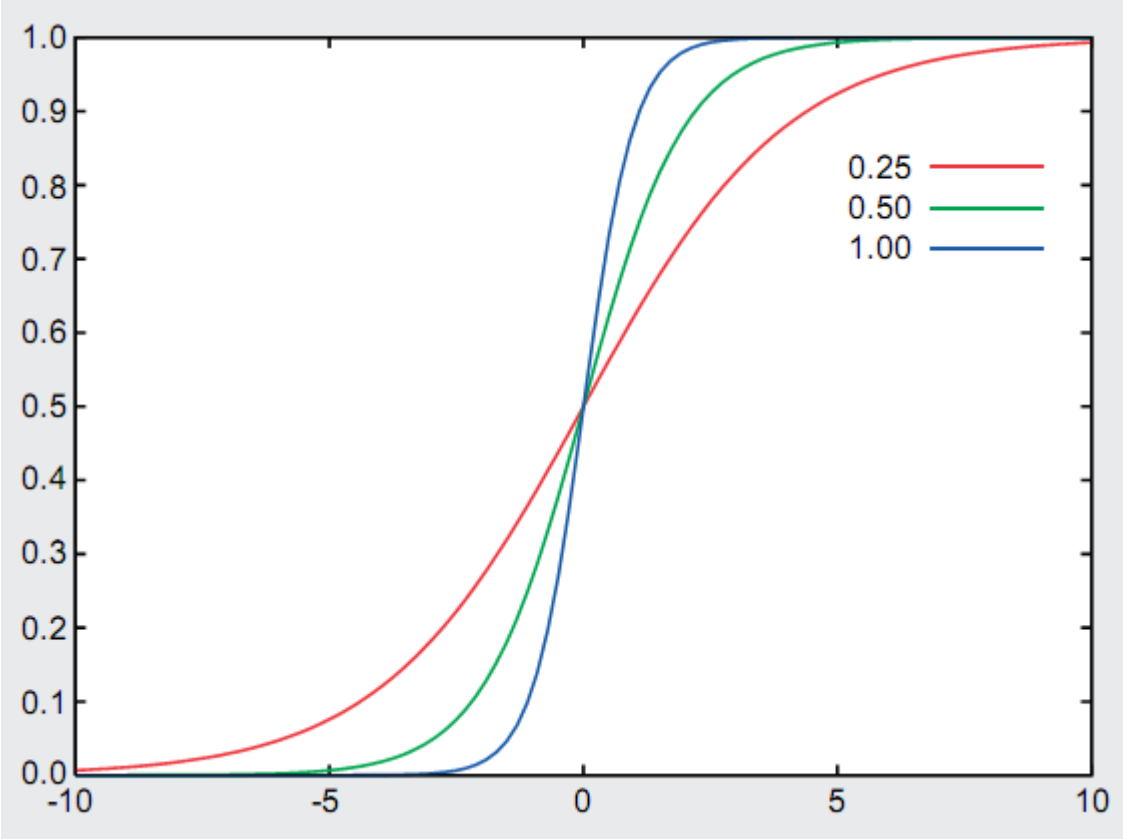
Aslında FANN kütüphanesi 5 katmanlıdan fazlası için de destek veriyor. Fakat pratikte pek faydalı olmadığından kullanmadık yani FannTool ile En fazla 3 saklı katman tanımlayabilirsiniz. Ve dolayısıyla Katman Sayısın en çok 5 olur.

**Training Method** : Eğitim Metodları FANN kütüphanesinde halen 4 çeşit algoritma vardır bunlar

1. FANN_TRAIN_INCREMENTAL : standart Geri yayılma algoritmasıdır. Her eğitim verisi sonunda ağırlıklar güncellenir.

2. FANN_TRAIN_BATCH : standart Geri yayılma algoritmasıdır. Bütün eğitim verileri uygulanıp MSE hesaplandıktan sonra ağırlıklar güncellenir.
3. FANN_TRAIN_RPROP : batch training metodunun çok daha gelişmişidir, özelliği gereği öğrenme hızını kullanmaz. Kendi ayar parametrelerinin değiştirilmesi de metodun işleyişini bilmeyen kullanıcılara tavsiye edilmez. Bu metod ilk olarak 1993 yılında Riedmiller ve Braun, tarafından geliştirilmiştir. FANN da kullanılan iRPROP- ise Igel ve Husken, tarafından – 2000 - geliştirilmiş bir RPROP çeşididir
4. FANN_TRAIN_QUICKPROP : Fahlman, tarafından 1988 geliştirilmiş bir eğitim metodudur.

**Activation Function Hidden** : Saklı Katman Eşik Fonksiyonu
**Activation Function Output** : Çıkış Katmanı Eşik Fonksiyonu




Yukardaki grafik Sigmoid Fonksiyonuna aitdir. Fine Tuning kısmında göreceğiniz "Stepness" seçeneklerinin de ne ifade ettiğini görebilirsiniz grafikte Sigmoid fonksiyonu değişik stepness değerleriyle çizilmiş. Peki bu fonksiyonlar temelde doğrusallıktan kurtarır ve verileri bir sonraki katmana taşırken yine belli sınırlar içinde tutar. Daha detaylı bilgi için buraya bakın

Bu kısımlarda gereken ayarları kendiniz seçerek yapabileceğiniz gibi. Menüden
Neural Network -> Detect -> Optimum Trainin Algorithm

Neural Network -> Detect -> Optimum Activation Functions
Seçenekleriyle FannTooldan Sizin için denemeler yapıp sonuça ulaşmasını isteyebilirsiniz.

**Train** : Eğitim iki şekilde yapma imkanına sahipsiniz.
**Train -> Normal** : Sabit Geometrideki ( katmanlar ve hücre -nöron- sayıları ) Yapay Sini Ağı eğitim işini yapar. Belirleyeceğiniz kritere ulaşıncaya Ya da belirttiğiniz adım sayısına gelinceye dek ağırlıklar değişmeye devam eder,

**Train -> Cascade** : Geometrik değişim ile yapılan eğitimdir. Boş bir YSA ile başlayıp nöron ekleyerek ve seçeneklerinini değiştirerek eğitir. Belirleyeceğiniz kritere ulaşıncaya Ya da belirttiğiniz hücre sayısı tamamalanıncaya kadar ağırlıklar değişmeye devam eder,

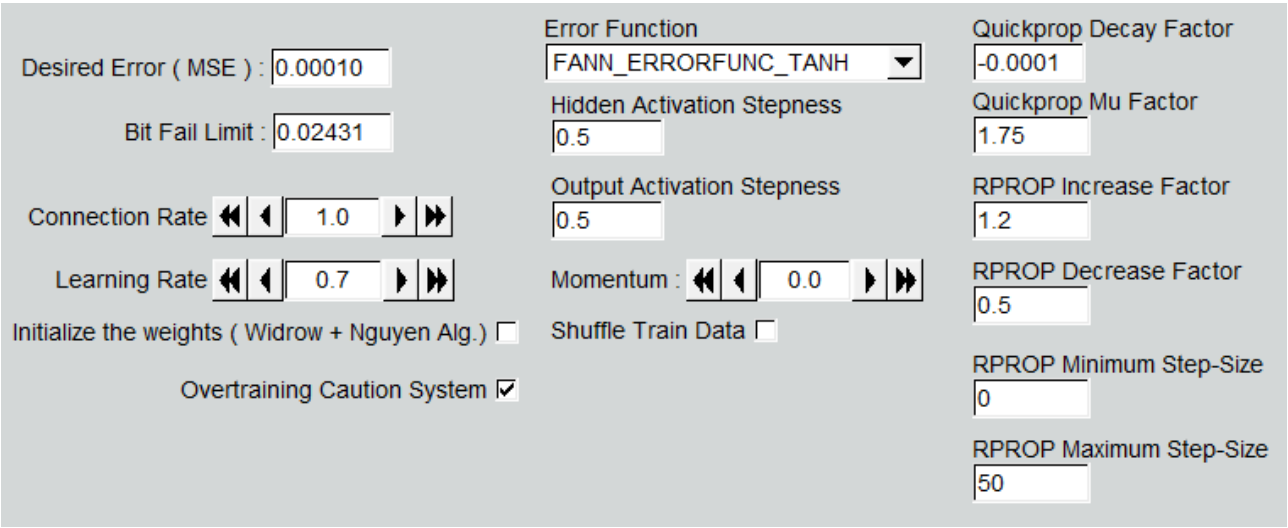
**Stop Function** : Eğtim işleminin durdurmak için gereken kriter Genelde MSE yani Ortalama Karesel Hata olarak seçilir. Diğer seçenek de Bit Fail Limt dir. "Fine Tuning" Kısmında Durdurma sevyelerini ayarlayabilirsiniz.

**Epoch Between Reports** : FannTool YSA'nın eğitimi esnasında size gidişatı grafik ve yazıyla gösterir. Grafik için Graphic sekmesi yazılı görmek için Log sekmesine bakmanız yeterli. Bu işlemin kaç adımda bir yapılacağını buradan ayarlayabilirsiniz. Eğer çok büyük verilerle çalışıyorsanız, bu değeri düşürmenizi tavsiye ederim.

**Max Epoch** : Eğitim işleminin en fazla kaç adım devam edeceğini belirliyorsunuz.

FannTool'un alt kısmındaki sekmelere de bakalım.

**FineTuning** :




Bu kısımda Eğitimi etkileyecek pek çok değişkene ince ayar yapma imkanı sunulur.

"*Quickprop Decay Factor*" , "*Quickprop Mu Factor*" Quickprop Eğitim metodu için geçerli olan değişkenlerdir.
"*RPROP Increase Factor* ", "*RPROP Decrease Factor*", "*RPROP Minimum Step-Size*", "*RPROP Maximum Step-Size*" RPROP Eğitim metodu için geçerli olan değişkenlerdir. Detayları için FANN dökümantasyonuna bakılabilir.
"*Desired Error ( MSE )*" ve "*Bit Fail Limit*" ile ilgili açıklamalar yukarda Stop Function kısmında anlatıldı.
"*Hidden Activation Stepness* " ve "*Output Activation Stepness* " ile ilgili açıklamalar Activation Function kısmında anlatıldı.

"*Learning Rate*" : Öğrenme hızı,  RPROP hariç diğer eğitim metodlarında geçerlidir.

"*Momentum* "  : Momentum

"*Connection Rate*" : 1 olduğunda tamamen bağlı bir YSA yapay kurar. Değerin küçüklüğüne göre bağlantı sayısını azaltır.

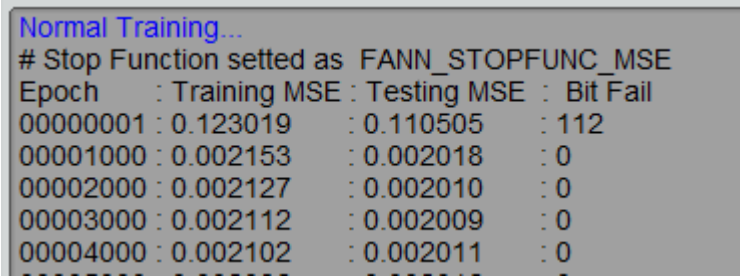
"*Shuffle Train Data*" : Eğtimim için kullanılacak verilerin sıralamasını rastgele olarak karışırtır.

"*Initialize the weights ( Widrow + Nguyen Alg.)*" : Başlangıçta YSAnın ağırlıkları rastgele seçilir. Bu seçenekle Derrick Nguyen ve Bernard Widrow bulduğu ve eğitimin daha hızlı olmasına yardımcı olduğu söylenen bir metod ile seçilir.

"Error Function" : hata hesap fonkisyonu iki seçeneğimiz var. birincisi standart lineer hata fonksiyonu diğeri tanh bir fonksiyon

"*Overtraining Caution System*" : Kısaltılmış haliyle OCS yani Aşırı Öğrenme Uyarı sistemi Yapay Sinir Ağlarının eğitimi süresince Hatanın istenen değerin altına düşürülmesine çalışılır. İşlem sonrasında Eğitim için kullanılmayan Verileriyle kontrol edilmelidir. Eğtim sonun da çok düşük hata değerine ulaşılmasına rağmen Test sonucunda çok yüksek hata değerleri çıkıyorsa buna Overtraining diyorlar ve sonuçta ulaştığımız YSA işimize yaramıyor çünkü sadece eğitim verisi için geçerli sonuçlar verebiliyor. Eğer Test verinizide yüklemişseniz ve Bu seçeneği aktif hale getirmişseniz. FannTool size eğitim sırasında Test verilerinin hata değerlerinide verir. Grafik olarakda Eğitim ve Test verilerinin hatalarının ortalaması kullanılarak çizer. Bu sayede siz Eğtiminizin ne yöne gittiğini görme şansına sahip olursunuz.

**Log** :




Yapılan her işlem sonrasında kullanıcıyı bilgilendirmek için yazılan mesajlar. Gerktiğinde

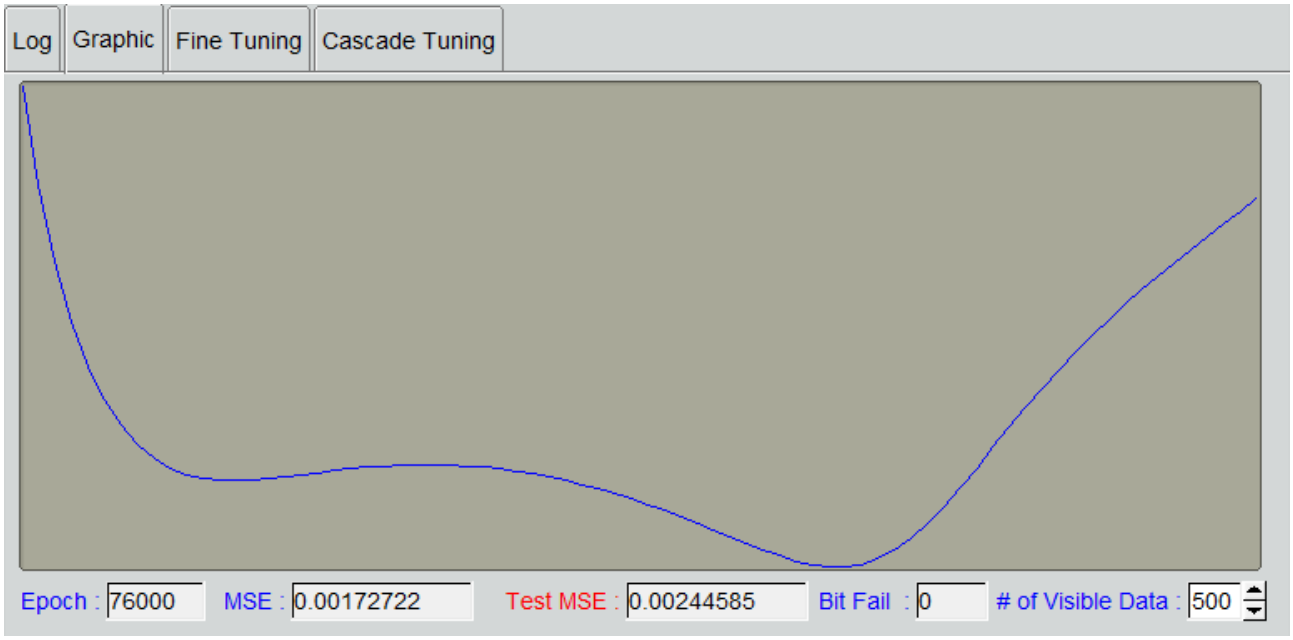
"Save Log"  ile kaydedebilir,

Ya da "Open Log" ile daha önce kaydettiğinizi yükleyebilir.

"Clear Log " ile temizleyebilirsiniz.

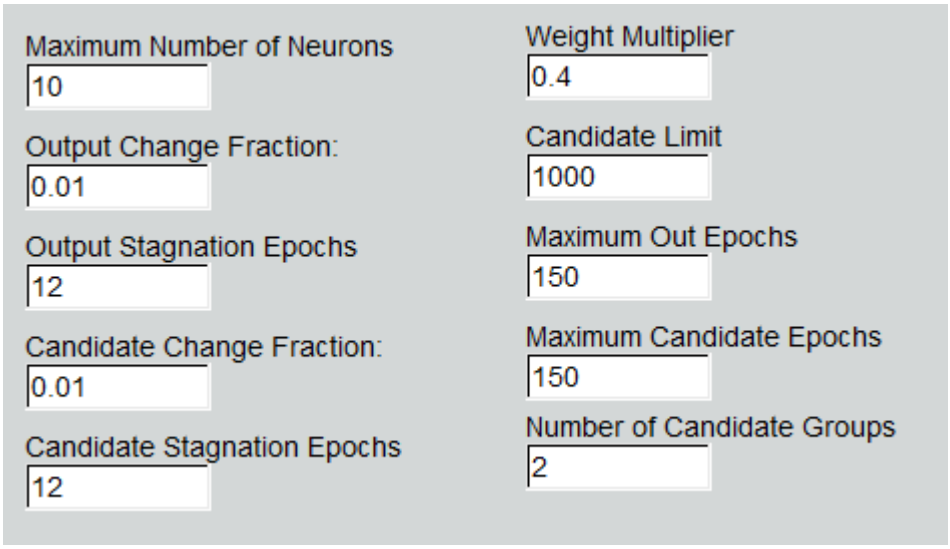
Bir sonraki işlem bir önceki Log u siler.  Mesela vakit alan Optimum Eğitim metodu Ya da  Eşik fonksiyonu tespitinden sonra Log u kayederek. Sonradan kullanabilirsiniz.

**Graphic** :




Eğitim esnasında Hata değerlerini grafik olarak izleyeceğiniz kısım. Eğer OCS seçeneğini işaretlemişseniz altta Test MSE de gösterilir çizilen grafikde Eğitim MSE ve Test MSE değerlerinin ortalmasının grafiği çizilir. OCS kapalıysa saadece Eğitim MSE nin grafiği çizilir.

**Cascade Tuning** :

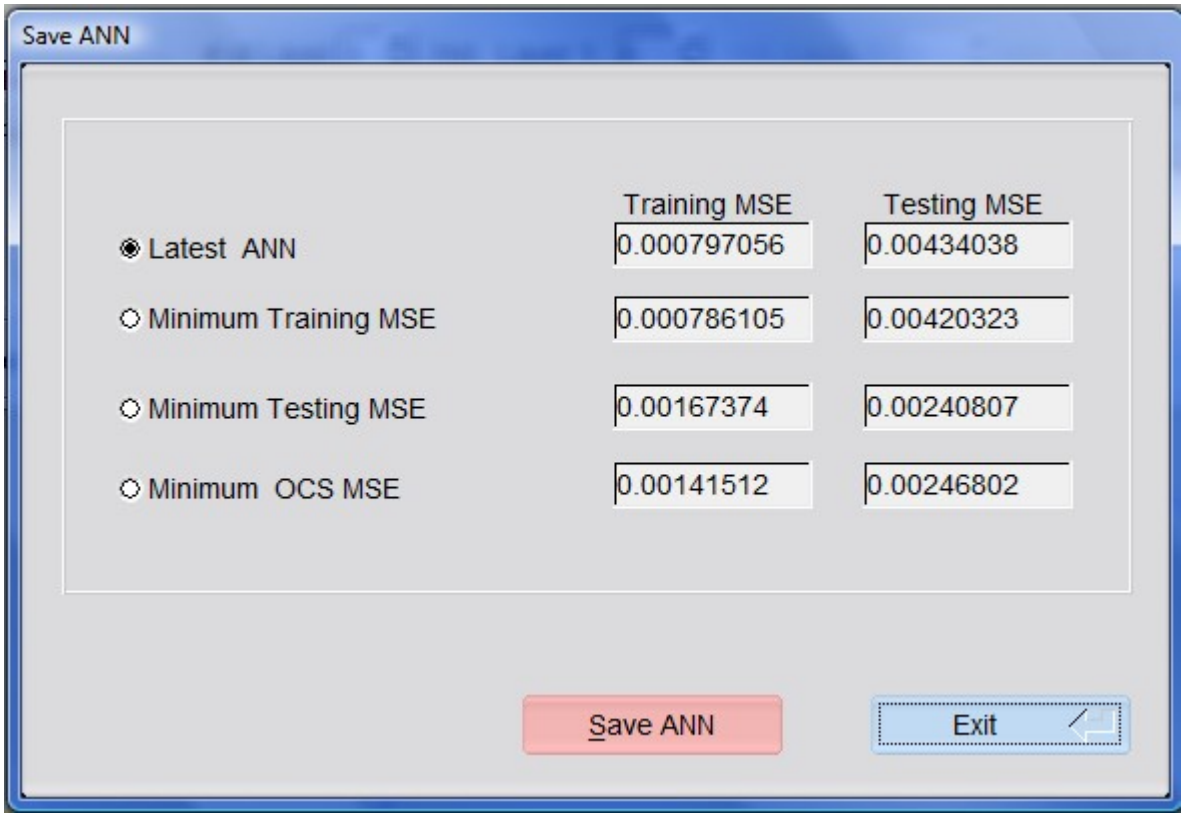



Fann kütüphanesinin bir imkanıdır Cascade. Boş bir YSA ile başlayıp nöron ekleyerek ve seçeneklerinini değiştirerek eğitir. Burada başlıca kullanacğımız parametre “Maximum Number of Neurons” dur. Bu eğitim sırasında eklenecek maksimum nöron sayısını belirler. Diğer parametreleri FANN'ın dökümantasyonundan öğrenebilirsiniz.

**YSA Kayıt** :

Eğitim işi bittiğinde yada

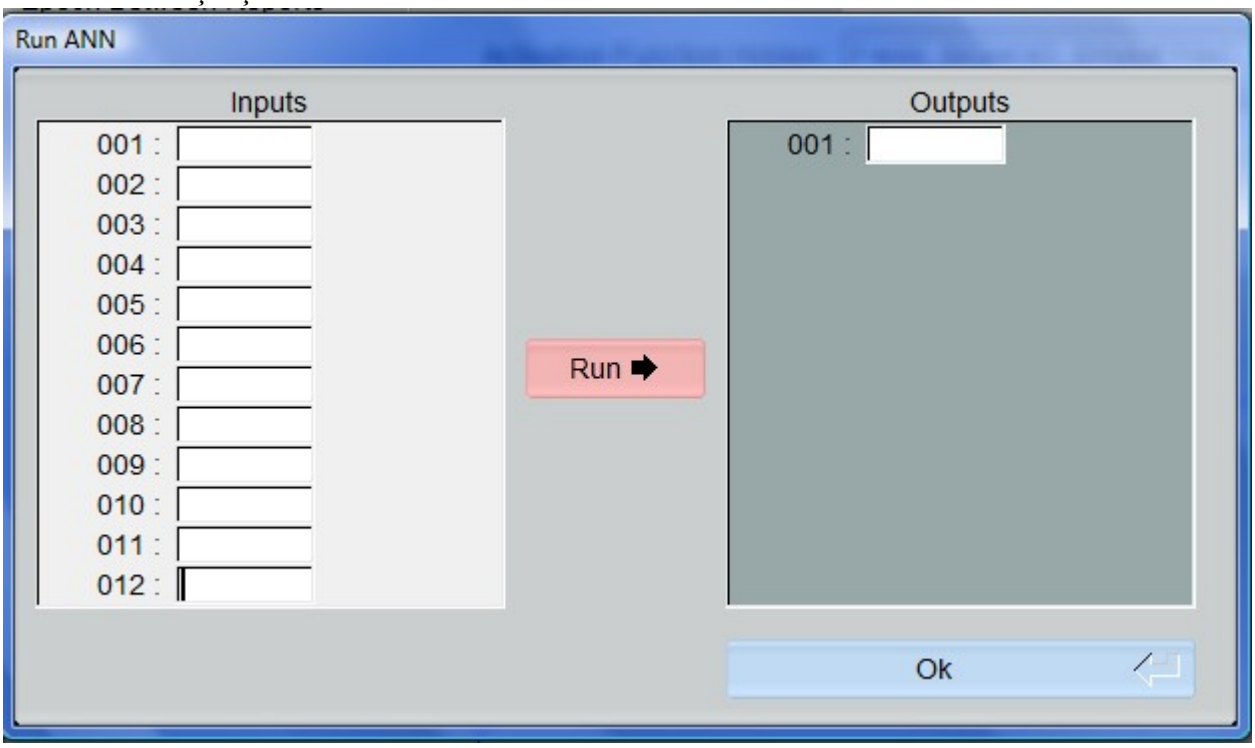
Stop ile işlemi durduğunuzda Yapay Sinir Ağını kaydeteme imaknına sahip olursunuz.



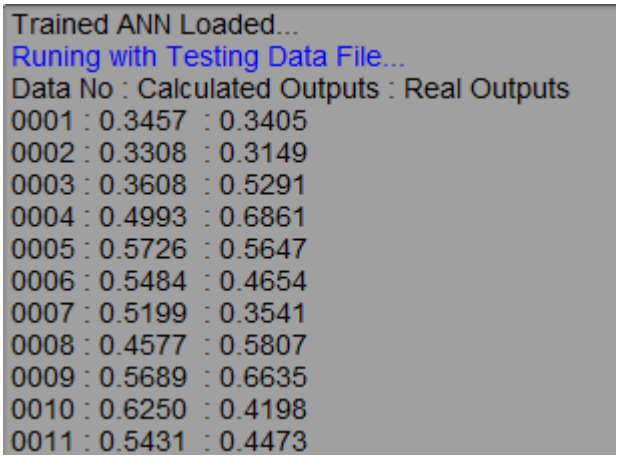

Ve karşınıza *"Save ANN"* penceresi gelir. Kaydetmek istediğiniz YSA yı seçip Save ANN diyerek kaydedebilirsiniz. Eğitim süresince grafikden de takip edebileceğiniz gibi MSE değerleri değişime uğrar. FannTool bu esnada minumum değerlere sahip yerlerde YSA'yı hafızasına alır. Kullanıcının isteğine göre de kaydeder.

- *Latest ANN* : Bu işlemin durduğu andaki YSA kayıt edilir.
- *Minimum Training MSE* : İşlem boyunce Eğitim Verisiyle ulaşılan en düşük hata anındaki YSA kayıt edilir.
- *Minimum Testing MSE* : İşlem boyunce Test Verisiyle ulaşılan en düşük hata anındaki YSA kayıt edilir. OCS aktif edildiğinde bu seçenekde aktif olur.
- *Minimum OCS MSE* : İşlem boyunce Test ve Eğitim verisiyle ulaşılan hata değerlerin ortalmasının en düşük olduğu anındaki YSA kayıt edilir. OCS aktif edildiğinde bu seçenekde aktif olur.

**Run** : Eğittiğimiz YSA' yı FANN kütüphaesi ile Program yazarak çalıştırabeleceğiniz gibi FannTool ile de çalıştıra bilirsiniz.



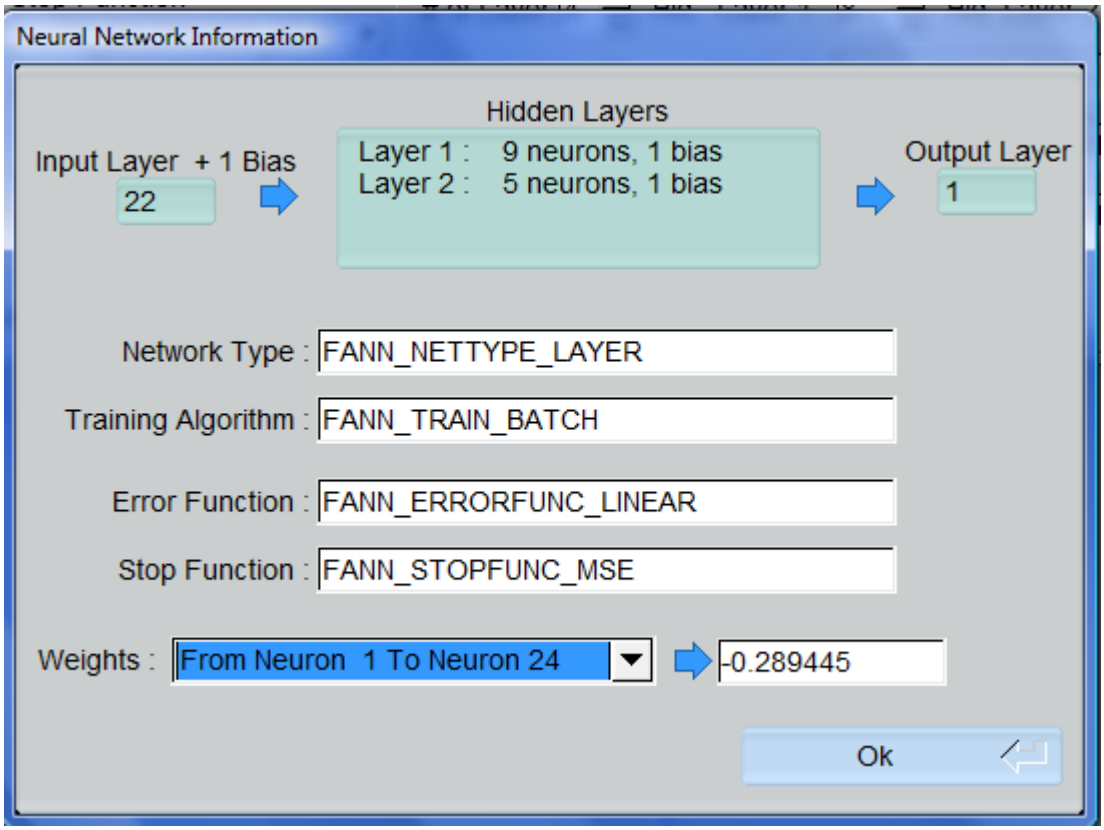

Run -> Normal : Seçtiğiniz YSA 'yı Inputs kısmında gireceğiniz verilerle çalıştırır ve sonucu outputs kısmına yazar. Dikkate edilecek şey ; Giriş verilerinin Ölçeklendirilmiş olarak vermeniz gerek, sonuçta çıkan sonuçda ölçeklendirilmiş olduğundan. Gerçek sonuç için çevirme yapmanız gerekir.



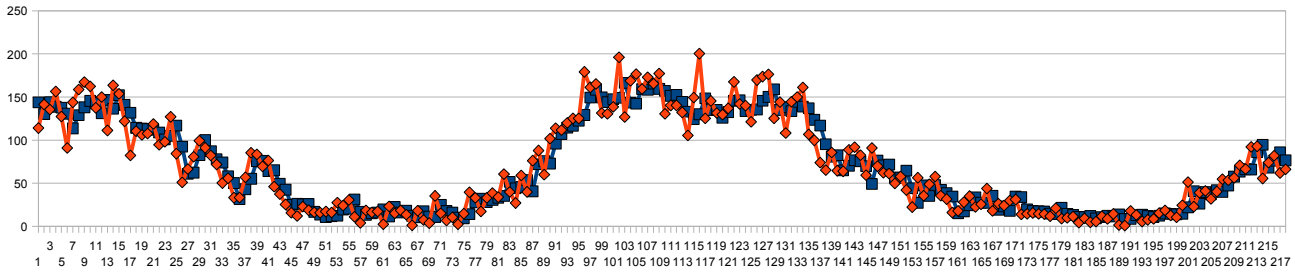

Run -> with File : Dosyadan çalıştırma . Bir gurup veriyle YSA yı çalıştırıp sonuçları karşılaştırmak istiyorsanız. Verileri FANN veri formatında hazırlayıp kaydedin bu dosyayı Testing Data File Kısmına yükleyin ve Run with File ile YSA'nızı seçtiğinizde. FannTool bu verilerle YSAnızı çalıştırır ve log penecersinde
veri no : Hesaplanan Sonuç : Gerçek sonuç
şeklinde yazar. Buradan Log'u kaydedip kullanabilirsiniz.

**Neural Network Information** :

Kayıt ettiğiniz YSA'nın  parametreleri yapısı ve ağırlıklarını görmenizisağlayan bir kısımdır.




Örneğimize dönersek, veri işleme kısmıyla ürettiğimiz eğitim ve test verileri “SunSpots-train.dat” ve “SunSpots-test.dat”  ile ve değişik parametreler kullanarak çeşitli eğitim denemeleri yaptık sonuç ta uygun olduğuna inandığımız YSA yı kaydettik “SunSpots-OCS.net” ve test verileriyle çalıştırdık. Sonuçta çıkan değerleri “SunSpotsRun.log” isimli metin dosyasına kaydettik. Sonuçları grafik olarak gösterebilmek için tablolama programına aktardık.”SunSpots.xls” ve bu sonuca ulaştık.




Yukarda bahsettiğimiz dosyalar bu dökümanla birlikte dağıtılıyor. Konuyu pekiştirmek için pratik yapmanızı öneririz. Yine Sitemizde başka örnek uygulamalarda var onların verilerinide inceleyebilirsiniz.

Sitemizde ulaşabileceğiniz diğer YSA uygulmaları

- Harf Tanıma
- Küresel Grafitli Dökme Demirlerde Nodül Sayısı tespiti
- Yapay Zeka Tıbbın Hizmetinde 1 ve 2
- Göz Bulma I ve II
- TekSatır Kod Yazmadan YSA
- YSA ile Kan Bağışı Tahmini
- FANN ile ilgili tüm Yazılar

## Sonsöz :

Yapay Sinir Ağlarının uygulama alanı çok geniş fakat nadiren kullanılıyor. FannTool programıyla uygulama işlemini kolaylaştırma çabasındayız. Bu kılavuz ile de bu programı nasıl kullanabileceğinizi anlatmaya gayret ettik. Eğer anlaşılmayan bir kısım, hata veya eksiklik varsa bize bildirirseniz seviniriz.

Konuyla ilgili problem ve sorular varsa bize mail adresimizden çekinmeden ulaşabilirsiniz, yardımcı olmakdan memnuniyet duyarız. Son olarak bu konuda çalışma yapmak isteyenlere bir nebze faydamız olabilirse ne mutlu bize...

BlueKid