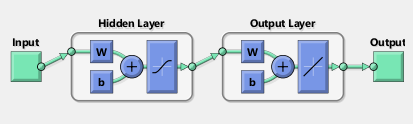
**The Neural Fitting App(Nöral Uydurma Uygulaması) (Giriş/Çıkış ve Eğri Uydurma)**

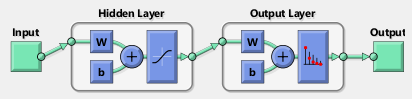
Uydurma problemlerinde, bir sinir ağının sayısal girişlerin bir veri kümesi ile sayısal hedeflerin bir kümesi arasında eşleşmesini istiyorsunuz. Bu problemin örnekleri arasında yakıt tüketimi ve hız ölçümlerine dayanarak motor emisyon seviyelerinin tahmin edilmesi (motor\_dataset) veya vücut ölçümlerine dayanarak hastanın bodyfat seviyesinin tahmin edilmesi (bodyfat\_dataset) bulunur. Nöral Uydurma uygulaması veri seçmenize, bir ağ oluşturmanıza ve eğitmenize ve ortalama kare hata ve regresyon analizini kullanarak performansını değerlendirmenize yardımcı olur.



Sigmoid gizli nöronları ve doğrusal çıkış nöronlarını (fitnet) içeren iki katmanlı bir ileri besleme ağı, tutarlı veri ve gizli katmanında yeterli nöronlar göz önüne alındığında çok boyutlu haritalama sorunlarına keyfi bir şekilde iyi uydurabilir. Ağ, yeterli bellek olmadıkça ölçeklendirilmiş eşlenik gradyan geri yayılımı (trainscg) kullanılacaksa, Levenberg-Marquardt backpropagation algoritması (trainlm) ile eğitilecektir.

**Pattern Recognition App(Desen Tanıma Uygulaması)(Desen Tanıma ve Sınıflandırma)**

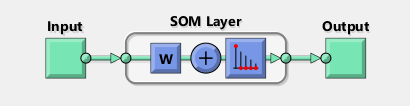
Örüntü tanıma problemlerinde, bir sinir ağının girdileri bir hedef kategoriler kümesinde sınıflandırmasını istersiniz. Örneğin, kimyasal analize dayalı (wine\_dataset); veya bir hücreyi, hücre büyüklüğünün düzgünlüğüne, küme kalınlığına, mitoz (kanser veri kümesi) durumuna bağlı olarak benign veya malign olarak sınıflandırır. Sinir Örüntü Tanıma uygulaması, verileri seçmenize, bir ağ oluşturmanıza ve eğitmenize ve çapraz entropi ve karışıklık matrislerini kullanarak performansını değerlendirmenize yardımcı olur.



Sigmoid gizli ve softmax çıkış nöronlarına (patternnet) sahip iki katmanlı ileri besleme ağı, gizli katmanında yeterli nöron verildiğinde vektörleri keyfi olarak iyi bir şekilde sınıflandırabilir. Ağ, ölçekli eşlenik gradyan geri yayılımı (trainscg) ile eğitilecektir.

**Clustering App(Kümeleme Uygulaması)**

Kümelenme problemlerinde, sinir ağlarının verileri benzer şekilde gruplandırmasını istiyorsunuz. Örneğin: insanları satın alma alışkanlıklarına göre gruplayarak yapılan pazar bölümlemesi; veri madenciliği, verileri ilgili alt gruplara bölerek yapılabilir; veya ilgili ifade modelleriyle birlikte genlerin gruplanması gibi biyoinformatik analizler. Sinir Kümelemesi uygulaması, çeşitli görselleştirme araçlarını kullanarak veri seçmenize, bir ağ kurmanıza ve eğitmenize ve performansını değerlendirmenize yardımcı olacaktır.



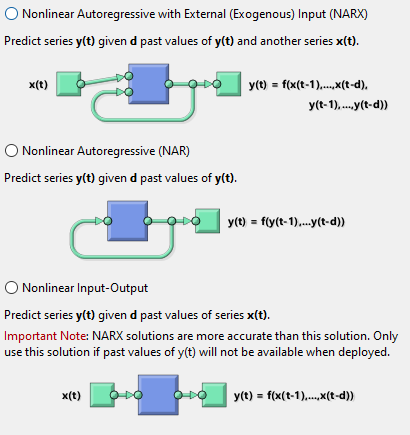
Kendi kendini düzenleyen bir harita (selforgmap), katman nöronlarının olduğu kadar herhangi bir sayıda vektör içeren bir veri kümesini herhangi bir sayıda boyutta sınıflandırabilen rekabetçi bir katmandan oluşur. Nöronlar, tabakanın dağılımın bir gösterimini ve veri setinin topolojisinin iki boyutlu bir yaklaşımını oluşturmasını sağlayan bir 2D topolojisinde düzenlenir. Ağ, SOM parti algoritması (trainbu, öğrenen) ile eğitilmiştir.

**Time Series(Dinamik Zaman Serileri)**

Tahmin, gelecekteki değerleri tahmin etmek için bir veya daha fazla zaman serisinin geçmiş değerlerinin kullanıldığı bir tür dinamik filtrelemedir. Kılavuzlanmış gecikme çizgileri içeren dinamik sinir ağları, doğrusal olmayan filtreleme ve tahmin için kullanılır.

Tahmin için birçok uygulama var. Örneğin, bir finansal analist bir hisse senedinin, tahvilin veya başka bir finansal aracın gelecekteki değerini tahmin etmek isteyebilir. Bir mühendis bir jet motorunun olası arızasını tahmin etmek isteyebilir.

Tahmini modeller ayrıca fiziksel sistemlerin dinamik modellerini oluşturduğunuz sistem tanımlaması (veya dinamik modelleme) için de kullanılır. Bu dinamik modeller, üretim sistemleri, kimyasal işlemler, robotik ve havacılık sistemleri de dahil olmak üzere çeşitli sistemlerin analizi, simülasyonu, izlenmesi ve kontrolü için önemlidir.

Bu araç, gösterilen üç tür doğrusal olmayan zaman serisi problemini çözmenizi sağlar. 

**Levenberg-Marquard Algoritması**

Levenberg Marquardt temel olarak maksimum komşuluk üzerine kurulmuş en az kareler hesaplama metodudur. Bu algoritma Gauss-Newton ve gradient-descent algoritmalarının en iyi özelliklerinden oluşur ve bu iki metodun kısıtlamalarını kaldırır. Genel olarak bu metot yavaş yakınsama probleminden etkilenmez. Gauss-Newton algoritmasının kombinasyonu olan Levenberg-Marquart algoritması optimizasyon problemlerinde eğimli düşüş algoritmasına göre (gradient descent) daha etkindir. Ayrıca daha hızlıdır

Dik iniş (steepest descent) ve Newton algoritmalarından türetilen LM algoritması güncellemesi (1)’de verilmiştir.

Levenberg-Marquardt (LM) algoritması yapay sinir ağlarının eğitiminde sağlamış olduğu hız ve kararlılık nedeni ile tercih edilmektedir. Örnek olarak **:** Rezidüel Anomaliden Derinlik ve Yarıçap Parametrelerinin Bulunması



w ağırlık vektörü, I birim matris, µ kombinasyon katsayısıdır. J (PxM)xN boyutunda Jacobian matrisini, e (PxM)x1 boyutunda hata vektörünü göstermektedir. P eğitim örnek sayısını, M çıkış sayısını ve N ağırlık sayısını göstermektedir. µ ayarlanabilir bir parametredir. Eğer bu parametre çok büyükse yöntem dik iniş metodu gibi çok küçükse Newton metodu gibi davranmaktadır. Bu parametre için uyarlamalı bir yapı (2)'de verilmiştir.



Denklem (2)’de k sabit bir sayıdır. ‘E’ uygunluk değerini göstermektedir.

# Scaled Conjugate Gradient (SCG)

# Sinir ağları için denetimli bir öğrenme algoritmasıdır ve eşlenik gradyan metotları sınıfının bir üyesidir.

SCG'nin standart backpropagation(geri yayılım) gibi diğer algoritmalarla karşılaştırılmasında epochs(devir) sayısı önemli değildir. Aslında, SCG'deki bir yineleme iki degradenin(indirgeme, gerileme) hesaplanmasını gerektirir ve bir tanesi hata fonksiyonuna çağrı yaparken, standart ters yayılımdaki bir yinelemenin bir degradenin hesaplanmasına ve bir hata fonksiyonuna bir çağrının yapılmasına ihtiyaç duyar. Møller , eğitim setindeki tüm kalıplardan bir ileri geçişin karmaşıklığına eşdeğer bir *karmaşıklık birimi* (cu) tanımlar . O zaman hata hesaplamak degrade hesaplanırken 1 cu maliyeti 3 cu maliyeti tahmin edilebilir. Møller'in metriğine göre, bir SCG yinelemesi standart geri yayılım yinelemeleri kadar karmaşıktır .

**Not:** SCG'nin SNNS uygulaması çok iyi optimize edilmediğinden, CPU zamanı mutlaka iyi bir karşılaştırma kriteri değildir.

**Bayesian Regularization Algoritması**

Bayesian Regulation Method: Bayesian regülâsyonu Levenberg Marquardt optimizasyonuna göre ağırlık ve bias değerlerini günceller. Karesel hata ve ağırlıkların kombinasyonu minimize eder ve ağı üretmek için doğru kombinasyonu belirler. Daha iyi genelleştirme yeteneğine sahip bir ağ oluşturmak için Mackay [11] regülâsyonla ağ parametrelerinin boyutunu sınırlayan yöntem önermiştir. Regülâsyon teknikleri ağırlık değerlerinin daha küçük değerlerde kalması için ağı zorlar. Bu ağın cevabının daha yumuşak olmasına, ağın ezberleme (over fitting) olasılığının azalmasına ve gürültüyü yakalamasına neden olur.

Bayesian düzenli yapay sinir ağları (BRANN'ler) standart geri yayılma ağlarından daha dayanıklıdır ve uzun çapraz doğrulama ihtiyacını azaltabilir veya ortadan kaldırabilir. Bayes düzenlileşmesi, doğrusal olmayan bir regresyonu, bir sırt regresyonu şeklinde "iyi bilinen" bir istatistiksel soruna dönüştüren matematiksel bir süreçtir. BRANN'lerin avantajı, modellerin sağlam olması ve geri yayılma gibi normal regresyon yöntemlerinde O (N2) olarak ölçeklendirilen doğrulama işleminin gereksiz olmasıdır. Bu ağlar, QSAR modellemesinde ortaya çıkan, model seçimi, modelin sağlamlığı, doğrulama kümesi seçimi, doğrulama çabasının boyutu ve ağ mimarisinin optimizasyonu gibi birçok soruna çözüm sunar.

Kanıt prosedürleri, eğitimi durdurmak için objektif bir Bayesian kriteri sağladığından. Bunların üstesinden gelmek de zordur, çünkü BRANN bir dizi etkili ağ parametresi veya ağırlığını hesaplar ve eğitir, konuyla ilgili olmayanları etkin bir şekilde kapatır. Bu etkili sayı genellikle standart bir tam bağlantılı arka yayılım sinir ağındaki ağırlık sayısından oldukça küçüktür. Girdi değişkenlerinin otomatik alaka düzeyi tespiti (ARD), BRANN'ler ile birlikte kullanılabilir ve bu, ağın her girdinin önemini "tahmin etmesini" sağlar. ARD yöntemi, modellemede kullanılan alakasız veya yüksek derecede ilişkili endekslerin, aktivite verilerinin modellenmesinde en önemli değişkenlerin hangisi olduğunu göstermenin yanı sıra ihmal edilmesini sağlar. Bu bölüm, BRANN yöntemini ve bir BRANN-QSAR modelini üretmek için bir akış çizelgesini tanımlayan denklemleri özetlemektedir. BRANN'lerin bir dizi veri setinde kullanılmasının bazı sonuçları gösterilmiş ve diğer doğrusal ve doğrusal olmayan modellerle karşılaştırılmıştır.