

Yıldız Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

Şekil Tanıma
Final Projesi

Selçuk BAŞAK
08501008

Not: Ödevi hazırlamak için geliştirdiğim uygulama ve kaynak kodları ektedir.

Giriş:

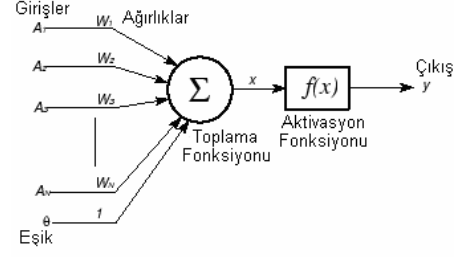
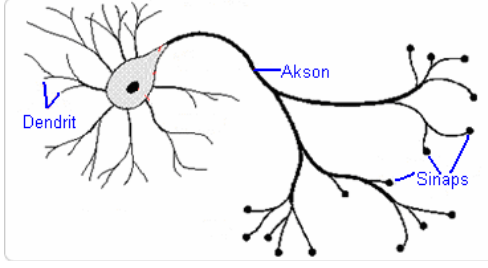
Bu Çalışmada, MNIST veritabanından alınmış el yazısı 0-9 arası sayılara ait örnekler Çok Katmanlı Yapay Nöron Ağlarının Back-Propagation (MLP) yöntemi ile sınıflandırılmıştır.

Sayı örneklerine ait dosya <http://yann.lecun.com/exdb/mnist/> adresindeki MNIST veritabanında bulunan el yazısı karakterlerden elde edilmiş 0-9 arası sayılardan oluşturulmuştur. Dosyadaki her satırdaki ilk 64 değer sayıya ait özellikleri(gri seviyesinde piksel değerleri), 65. değer ise sayının kaç olduğunu göstermektedir. Bu örneklerden her sayı için ilk 5'er örneği **eğitim** için geri kalan örnekleri ise **test** için kullanıldı. Eğitim örneklerini test için kullanılmadı.

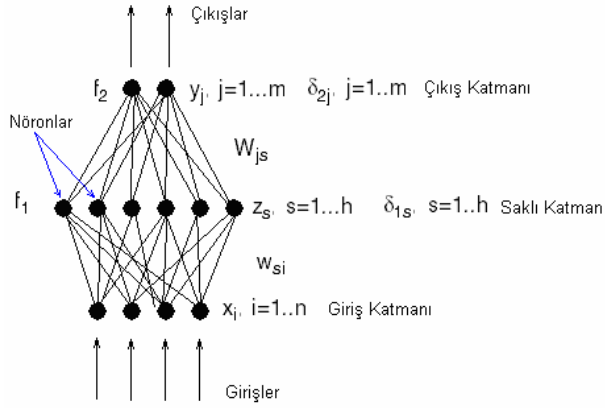
Eğitim için her rakam için 5 er tane olarak 50 örnek eğitim için kullanıldı. Geriye kalan 3773 örnek test için kullanılmıştır.

Yöntem

Yapay Nöron Ağları-Çok Katmanlı Perceptron(Multi Layer Perceptron-MLP) yöntemi, canlılardaki sinir hücrelerinden esinlenerek oluşturulmuştur. Gerçek ve yapay sinir hücreleri aşağıda gösterilmiştir.



$x = (\sum_{i=1}^n w_i * x_i) + \theta$, $y = \text{Signum}(x)$, $y = \tanh(x)$, $y = \text{Sign}(x)$ gibi aktivasyon fonksiyonları kullanılır. Yukarıda gösterilen tek bir yapay sinir hücresi perceptron olarak adlandırılır. Perceptron öğrenmeyi girişindeki ağırlıkları çıkışta oluşan hataya göre güncelleyerek yapar. Tek katmanlı bir yapay sinir ağı ile lineer olarak ayrılmış sınıflandırma yapılabilir. Ancak lineer olarak ayrılamayan sınıflandırma problemleri için çok katmanlı yapay sinir ağları- Multi Layer Perceptron kullanılmaktadır.



Şekilde gösterildiği gibi çok MLP giriş katmanı gizli katmana, gizli katmanlarda en son olarak çıkış katmanına bağlıdır. MLP Bağlantılar girişten çıkışa doğru olduğundan öğrenmek için back propogation algoritması kullanılır. Back propogation ile çıkış katmanında hesaplanan hata gizli katmana bağlantılardaki ağırlıklar oranında dağıtılarak yapılır.

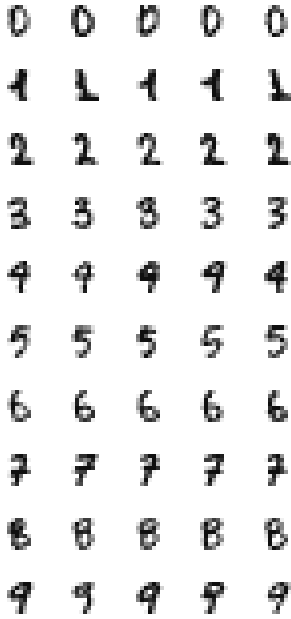
Uygulama

MLP yönteminin çalışması öğrenme katsayısı ve momentum değerleri sabit tutularak, gizli katmandaki düğüm sayısı değiştirilerek yapılmış ve aşağıdaki doru sınıflandırma başarısı aşağıda gösterilmiştir.

Gizli Katman Nöron Sayısı	Epoch	Eğitim Süresi(ms)	Tanıma Süresi (ms)	Accuracy
16	19733	38.641	31	65,49
32	13569	56.641	62	63,32
48	4969	29.453	94	66,18
64	3384	24.844	125	66,6
80	2898	26.687	141	64,7
96	2562	28.079	172	66,21
112	2340	30.016	203	69,02
128	2453	35.937	235	69,47
144	1946	31.922	250	63,61
160	1481	26.781	281	67,59
176	1895	37.781	312	67,74
192	1719	37.578	359	69,23
208	1345	31.843	375	69,15
224	1269	32.437	407	69,07
240	1254	34.219	438	64,62
256	998	28.860	469	66,02
272	1010	31.078	485	66,29
288	997	32.547	516	71,96
304	799	34.610	641	66,92
320	753	36.328	875	62,97
336	663	30.969	687	65,52
352	1355	70.219	922	67,08
368	601	30.235	734	67,98
384	590	30.735	766	62,34
400	605	32.906	796	64,38
416	564	36.453	922	65,81
432	609	40.407	954	70,55
448	481	30.344	1.094	64,27
464	418	25.828	922	65,28
480	525	33.297	937	69,04

Genel başarı oranları nispeten düşük bunun temel nedeni eğitim için örnek sayısı çok az olmasıdır. Bunun yanında elde edilen sonuçlardan görüldüğü gibi başarı ile gizli katmandaki nöron sayısı arasında düzgün orantılı bir değişim yok. Ancak 10-15'den az az gizli katman nöronu kullanılması durumunda eğitim yapılmasında networkün hatası azalamamakta ve eğitim tamamlanamaktadır. Eğer çok fazla kullanımının da başarıyı arttırmadığı ancak genel olarak eğitim tekrar sayısı (Epoch) azalmaktadır. Gizli katman nöron sayısı arttıkça sınıflandırma süresi de uzamaktadır. Gizli katman nöron sayısı değişimi eğitim süresinde düzenli bir değişim oluşturmamıştır. Ekteki sonuç dosyalarında her bir gizli katman nöron sayısı için detaylı sonuçlar vardır. En uygun gizli katman nöron sayısını belirlemek için cross validation yararlanmak uygun olacaktır.

Yapılan deęerlendirmede en yksek bařarının elde edildięi gizli katman nron sayısı 320 iin detaylı sonular ařaęıda gsterilmiřtir.

Training: Hidden Layer Node Count : 128 Learning Rate : 0,2 Momentum : 0,01 Min. Error : 0,0001 Training Epoch : 2453 Training Time : 35.937 ms. Train Set Performance: Correct Count : 50 Accuracy:%100 Incorrect Count: 0	Train Set Images 
Classification: Test Set Performance: Data Count : 3773 Correct Count : 2621 Incorrect Count: 1152 Mean Squared Error : 0,4581705471735 Accuracy : %69,47 Classification Time : 235 ms.	

		Predicated Digits										Correct	Incorrect	Accuracy
		0	1	2	3	4	5	6	7	8	9			
Actual Digits	0	365	0	0	2	2	0	1	0	1	0	365	6	98,38
	1	0	265	47	3	51	3	3	10	2	0	265	119	69,01
	2	15	0	340	5	0	0	0	13	2	0	340	35	90,67
	3	0	6	5	358	0	4	0	3	0	8	358	26	93,23
	4	14	44	9	9	167	4	77	50	4	4	167	215	43,72
	5	28	1	13	29	2	269	7	2	5	15	269	102	72,51
	6	2	20	2	0	1	0	345	1	0	1	345	27	92,74
	7	2	3	6	15	3	0	0	345	0	8	345	37	90,31
	8	4	31	60	62	23	1	18	23	114	39	114	261	30,40
	9	6	2	13	188	33	33	7	9	33	53	53	324	14,06

Yukarıdaki tabloda her bir resim iin elde edilen tanıma sonuları gsterilmiřtir. Bu sonuları eęitim iin kullanılan resimlerle beraber deęerlendirmek gerekir.

0,3 2,6 ve 7 %90'nın zerinde doęru tanıma bařarısı saęlanmış.

1 ve 5 iin yaklaşık %70 yani ortalama seviyede bir bařarı saęlanmış.

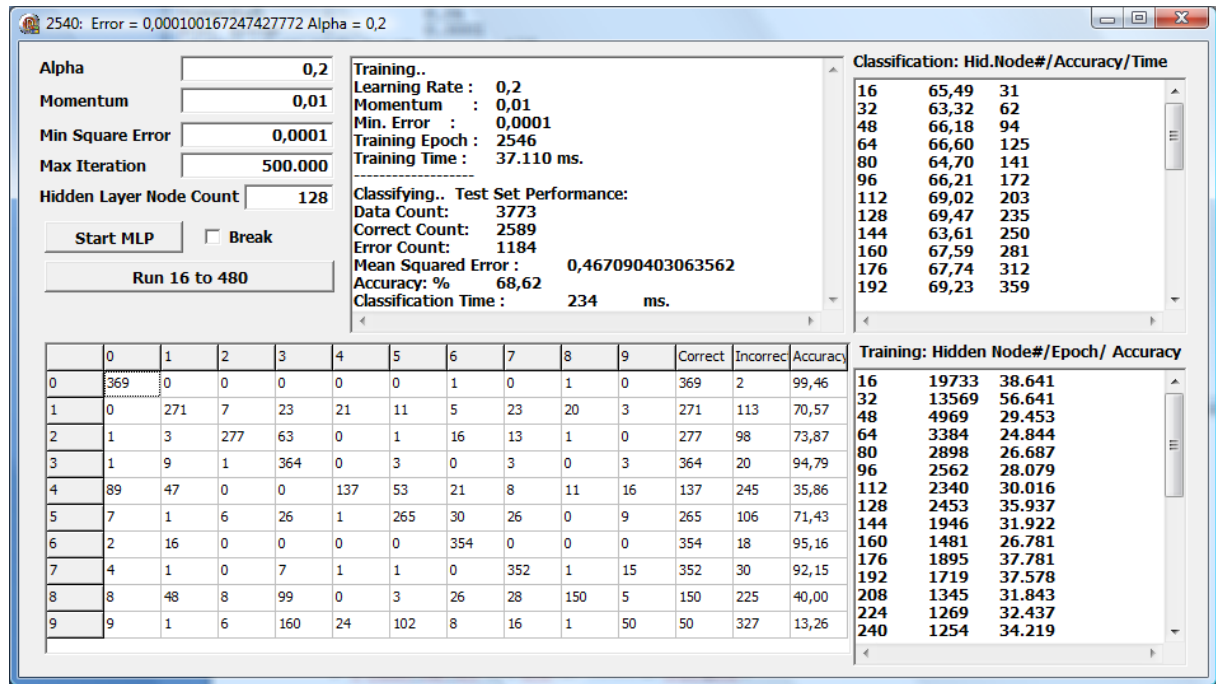
4,8 ve 9 iin ok dřk bir bařarı elde edilmiř.

Bu sonulara gre 0,3 2,6 ve 7 ok iyi ayırt edilebilir zelliklere sahip oldukları ve eęitim rneklerinin doęru seilmiř olduęunu dřnebiliriz.

4 ile 9 eęitim rnekleri bir birlerine ok benzemektedir. Muhtemelen bu nedenle eęitim sonucunda iyi bir sınıflandırma saęlamamıřtır. 8 ise řekli ile dięer tm rakamlara en ok benzemesi nedeniyle ky bir bařarı elde etmiř olabilir. 0-8 arası rakamlar en ok kendisi olarak tanınırken 9 iin en ok benzeme 3 ıktı. Bunun nedeni 3'n daha baskın olması ve 9 ile 3'n ok ortak řekil zellięi iermesi olabilir.

MLP Back Propagation Algoritması

1. Set weights [-0.5, 0.5] randomly
2. iter=0
3. mean_sqr_err =0
4. For each item in trainset
 - 4.1. get sample (inputs and desired outputs)
 - 4.2. evaluate hidden layer outputs
 - 4.3. evaluate output layer output
 - 4.3. calculate error between desired output and output layer outputs
 - 4.4. mean_sqr_err= mean_sqr_err+ (desired output - output)²
 - 4.5. clear hidden layer error
 - 4.6. calculate delta_weights of output layer
 - 4.7. calculate hidden layer weighted error add to hidden layer error
 - 4.8. calculate delta_weights of hidden layer
 - 4.9. update weights with delta_weights.
5. if mean_sqr_err>0.05 and iter<MAXITER then goto 3



Bu çalışmada rakam tanıma için 10 çıkışlı tek gizli katmalı MLP kullanıldı. Burada her bir çıkış aktif iken ilgili rakamı belirtmiştir. Bu yaklaşım yerine 2ⁿ şekilde 4 çıkış içinde çalışma yapıldığında başarı fazla bir etkisi görülmedi. Ancak binary kodlama ile 0-15 arası sonuçlar elde edilebildiğinden 0-9 arası rakamlar için 10 ile 15 arasında sonuçlarda alınmaktaydı. Bu nedenle binary kodlama yerine 10 tane çıkıştan en yüksek sonucu veren ilgili rakamı gösterecek şekilde kullanıldı. Bu çalışma için çok büyük süre farkı olmadı ancak çıkış nöron sayısının artması işlem süresini uzatacağı göz önüne alınarak daha çok çıkış gerektiren durumlarda binary kodlama tercih edilebilir.

Sonuç

Eđitim için kullanılan örneklerin çok az olması nedeniyle ortalama başarı düşük gözöktü. Eđer yeterli sayıda örnek ile eğitim yapılsaydı %90'lara yakın başarı elde edilebiliyor. Çok az eğitim örneđi kullanılmasına rağmen sistemin başarısı oldukça etkileyici buldum. Rakamların yarısını %90'dan yüksek oranla tanıdı. Bir birine benzemeyen rakamları çok iyi ayırt ederken benzeyenlerde problem yaşanıldıđı göröldü. Burada eğitim seti kısıtlı olduđundan çok sağlıklı yorum yapmak mümkün deđil. Ancak şekil olarak bir birine benzeyen rakamlardaki düşük başarı oranı, şeklin piksel özelliklerinin momenti alınıp veya PCA ile en iyi ayırt edici özellikler oluşturup, bu yeni özellikler üzerinden MLP uygulanması ile arttırılabilir.

*İşlem zamanları Delphi 2006 ile geliştirilen native win32 uygulaması ile, Vista 64 bit ile Intel Q9550 2.83 GHz Quad Core işlemci ve 800MHz DDR2 hafızalı bilgisayar üzerinde hesaplanmıştır.