

Yapay Sinir Ağları
Perceptron ve Delta Kuralı ile
Karakter Tanıma

Bilal Kahraman

17011062

Ders: Yapay Sinir Ağlarına Giriş

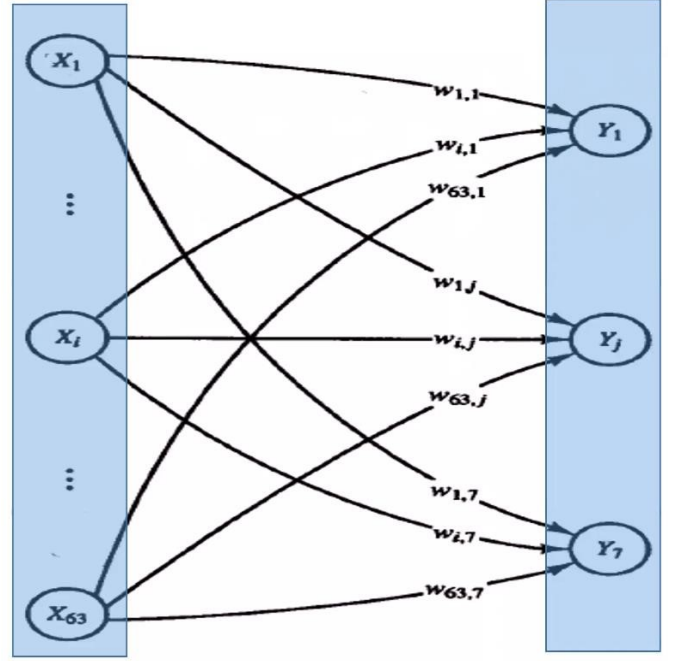
Dersi veren: Doç.Dr. Sırma ÇEKİRDEK YAVUZ

Kasım 22, 2020

Mimarinin Amacı ve Çalışma Mantığı

Toplamda 3'er farklı font ile 7 tane harf vardır. Her bir harfin bilgilerini içeren dosyada 63 tane veri vardır. Algoritma bu 63 tane veriyi okur ve her veriyi kendisiyle ilgili ağırlık ile çarpılarak toplanır. Bu toplama en son bias da eklenir ve toplam bir eşik değerinden geçirilir. Eğer bu toplamdan çıkan değer eşik değerinden büyük olursa nöron ateşlenir ve 1 değerini alır. Eğer küçük olursa nöron tepki vermez ve 0 değerini alır. Oluşan bu değer olması gereken hedef değeri ile karşılaştırılır. Eğer hedeflenen değer ile çıkan değer eşit değilse ağırlıklar ve bias istenilen kural ile güncellenir. Bu işlem %100 başarıya ulaşana kadar ya da epoch sayısı tamamlanana kadar devam eder.

Girdi olarak iki farklı tür vardır. Bunlar Binary ve Bipolar versiyonlardır. Binary versiyonda dosyanın içerisindeki veriler 1 ve 0 ile yazılmıştır. Bipolar da ise veriler 1 ve -1 halindedir.



W_{new} = yeni ağırlık

W_{old} = eski ağırlık

a = öğrenme katsayısı

input = girdi değeri

output = çıktı değeri

target = hedef değeri olmak üzere

R = Başarı Oranı

F = Girdi dosya sayısı (21)

T = Çıktı nöron sayısı (7)

D = Output ve target eşleşme sayısı

Perceptron Kuralı

Formülü: $W_{new} = W_{old} + a * input * target$

Delta Kuralı

Formülü: $W_{new} = W_{old} + a * input * (target - output)$

Şeklinde. İkisi de öğretmenli öğrenme modelidir ve öğrenme hata oldukça gerçekleşir.

Başarı oranı formülü: $R = 100 * D / (F * T)$, Eğer hiçbir nöron ateşlenmezse oran = %85.71

Her bir harf için hedeflenen çıktı değerleri tablodaki gibidir.

| | Nöron 1 | Nöron 2 | Nöron 3 | Nöron 4 | Nöron 5 | Nöron 6 | Nöron 7 |
|---|---------|---------|---------|---------|---------|---------|---------|
| A | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| J | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| K | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Birkaç tane program çıktısı örneği aşağıdaki gibidir.

| Öğrenme Katsayısı | Epoch Sayısı | Threshold (Eşik değeri) | Öğrenme Kuralı | Girdi Türü | Başarı Oranı (%) |
|-------------------|--------------|-------------------------|----------------|------------|------------------|
| 0.05 | 150 | 150 | Perceptron | Binary | 95.92 |
| 0.05 | 150 | 150 | Perceptron | Bipolar | 100 |
| 0.05 | 10000 | 150 | Perceptron | Binary | 95.92 |
| 0.05 | 150 | 150 | Delta | Binary | 100 |
| 0.5 | 50 | 150 | Delta | Bipolar | 100 |
| 0.001 | 150 | 50 | Delta | Bipolar | 85.71 |
| 0.05 | 150 | 50 | Perceptron | Binary | 86.39 |

Burada farklı renklerde işaretlenmiş alanlardaki değişim başarı oranlarını etkilemiştir. Sırasıyla;

Binary – Bipolar girdi türü

Binary girdi 1 ve 0'lardan; bipolar girdi ise 1 ve -1'lerden oluşur. Öğrenme kurallarına bakıldığı zaman ikisinde de girdiler çarpım olarak işleme dahil oluyor. İki input türündeki tek fark olan 0 ile -1 ve çarpımda farklı roller oynuyorlar. 0 ile çarpım sonucunda ağırlık değişmez ve yerinde kalırken -1 ile çarpım sonucunda negatif olarak eşik değerinden uzaklaşıyor. Bu sebeple diğer şartlar aynı tutulduğu zaman öğrenme daha hızlı gerçekleşiyor.

Öğrenme Kuralı Farkı

İki kural arasındaki tek fark, delta kuralında target değerinden output değerinin çıkarılarak çarpıma dahil edilmesidir. Burada ağırlık güncellemesi yapılırken delta kuralında output targeta ne denli uzak ise ağırlık o denli güncellenecektir. Perceptronda ise ağırlıkların güncellenme hızı sabittir. Bu sebeple diğer şartlar aynı tutulduğu zaman Delta kuralı uygulandığı ilk anda target ile output arasındaki fark fazla olacağından daha hızlı öğrenmeye başlayacaktır.

Epoch sayısı

Epoch sayısı yapılan iterasyon sayısını verir ve diğer şartlar aynı tutulduğu zaman, daha çok iterasyon başarı oranını artırır. Ayrıca tablodan da görüldüğü üzere epoch sayısı ciddi anlamda artmasına rağmen öğrenme oranı sabit kalmıştır bunun sebebi perceptronda eğer input 1 ise ağırlığın güncellenip, 0 ise sabit kalmasından kaynaklanır. Bir harfte aynı piksel değeri sıfır iken değer değişmeyebilir ancak bir başka harfte aynı piksel değeri 1 ise artmaya devam edecektir. Bu sebeple mimarı bir harfte doğru cevabı verirken değer harfte iki harfi birbirine karıştırabilir.

Öğrenme Katsayısı

Öğrenme katsayısı direkt olarak çarpıma katıldığı için yeni ağırlık değerinin hızlı veya yavaş olarak artması veya azalmasına sebep olur. Bu da öğrenmeyi direkt olarak etkiler.

Threshold (Eşik Değeri)

Threshold değeri nöronun ateşlenmesi için gelmesi gereken inputu ölçer. Eğer gelen input değeri thresholddan büyük ise nöron ateşlenir. Eğer binary input kullanılıyorsa thresholdu düşük tutma başarı oranını düşürecektir çünkü farklı inputlar için yapılan ağırlık güncellemesi sebebiyle 0 sonucu alması gereken piksel değeri kolay bir şekilde thresholdu geçecek ve 1 değerini alacaktır. Eğer yüksek bir değer seçilirse başarı oranını arttırabilir.

Girdi dosyaları bozulursa

Hem binary hem de bipolar türde aynı fonttaki harflerin piksel değerlerini değiştirdim. Perceptron kuralında binary input kullanıldığında yanlış sonuçlar gözlemlenirken bipolar inputta gözlenmiyor. Bunun sebebi bipolar yapıdaki diğer input dosyalarının istenilen değere götürmesinden kaynaklanıyor. 1 ile -1 girdi olursa birbirlerinden uzaklaştıkları için diğer fontlar durumu düzeltecektir ancak binaryde 1 uzaklaştırırken 0 hareket ettirmeyeceğinden dolayı öğrenmede hatalar olabilir. Delta kuralında ise target-output durumundan dolayı hata olması durumunda kural yardımcı oluyor ve ilerlemesine izin vermiyor. Bu durumda eğer 0 olması lazımken 1 değeri çıkarsa delta kuralı ağırlıkların -1 yönde ilerlemesini sağlayacaktır. Eksik piksel olması durumunda matriste boşluklar olacağından kod hata verir.

Öğrenme kuralı = 0.06

Threshold = 150

Epoch sayısı = 500

Harflerdeki bozulma oranı harf başına 5 digittir ve sırasıyla bir font bozulduğunda, iki font bozulduğunda ve üç fontun tamamı bozulduğu şeklinde ilerler.

| Öğrenme kuralı | İnput versiyonu | Bir font | İki font | Üç font |
|----------------|-----------------|------------------------------|--|--------------------------------|
| Perceptron | Binary | Font2(B-E,E-K) Font3(E,K) | Font1(B-K,D-K) Font2(B-E) Font3(B-D-E,E-K) | Font2(B-E-K) Font3(B-D-E-K) |
| Perceptron | Bipolar | - | - | - |
| Delta | Binary | - | - | - |
| Delta | Bipolar | - | - | - |

Öğrenme kuralı = 0.06

Threshold = 150

Epoch sayısı = 150

Harflerdeki bozulma oranı harf başına 10 digittir ve sırasıyla bir font bozulduğunda, iki font bozulduğunda ve üç fontun tamamı bozulduğu şeklinde ilerler.

| Öğrenme kuralı | İnput versiyonu | Bir font | İki font | Üç font |
|----------------|-----------------|----------------------------------|--------------------------------|----------------------------------|
| Perceptron | Binary | Font1(B-K,D-K,E-K) Font3(B-E) | Font1(B-K,D-J-K) Font3(B-E) | Font1(B-K,B-D-E-K) Font3(B-E) |
| Perceptron | Bipolar | - | - | - |
| Delta | Binary | - | - | - |
| Delta | Bipolar | - | - | - |

Burada da görüldüğü üzere bozulmalar perceptron öğrenme kuralında binary input kullanıldığı zaman gerçekleşiyor. Ancak buradaki piksel bozulmalarını elimle girdiğim için aynı bozuklukları aynı koordinatlara denk getirme ihtimalim yüksek olduğundan öğrenme kuralı bundan etkilenmemiş olabilir.

Kod ve Program

Kod yazımında C++ dili kullanıldı ve compiler bilgileri aşağıdaki gibidir;

```
C:\Users\HP\Documents\Visual Code\Artificial Neural Network>g++ --version
g++ (MinGW.org GCC-6.3.0-1) 6.3.0
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Program bütün parametreleri kullanıcıdan ister ve Kodun dosyaları açıp çalışabilmesi için .rar uzantılı dosyanın içerisindeki Filenames.txt dosyası ve Binary_Karakterler_Tam ile Karakterler_Tam dosyaları ile aynı dizinde bulunmalıdır. Filenames.txt dosyası açılacak dosyaların ismini tutar ve kullanıcının input tercihine göre klasörleri açarak okuma yapar. İlgili parametreleri alıp işlemi tamamladıktan sonra bütün harfleri mimariden geçirip cevabını ekrana çıktı olarak verir ve başarıyı oranını da altına ekler. Hesaplanan ağırlıkları ve girilen parametreleri weight_matrix.txt dosyasını oluşturur ve kaydeder.

Programdan birkaç çıktı;

```
Enter the learning rate(between 0-1): 0.005
Enter the epoch number: 9000
Threshold: 500
Press 1 for Perceptron rule
Press 2 for Delta Rule
1
Filesnames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
1
All .txt files is opened from Binary_Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 0 0 1
Success rate: 98.64
```

```
Enter the learning rate(between 0-1): 0.0015
Enter the epoch number: 12000
Threshold: 63
Press 1 for Perceptron rule
Press 2 for Delta Rule
2
Filesnames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
2
All .txt files is opened from Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
Success rate: 100
```

```

Enter the learning rate(between 0-1): 0.0015
Enter the epoch number: 12000
Threshold: 63
Press 1 for Perceptron rule
Press 2 for Delta Rule
1
Filenames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
1
All .txt files is opened from Binary_Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 1 1 0 1
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 1 0 1
1 0 0 0 0 0 0
0 1 0 1 1 0 1
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 1 1 0 1
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 1 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 1 0 1
Success rate: 89.12

```

```

Enter the learning rate(between 0-1): 0.091
Enter the epoch number: 650
Threshold: 50
Press 1 for Perceptron rule
Press 2 for Delta Rule
1
Filenames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
1
All .txt files is opened from Binary_Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 1 1 0 0
0 0 1 0 0 0 0
0 1 0 1 0 0 0
0 1 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 1 0 1
1 0 0 0 0 0 0
0 1 1 1 1 0 0
0 0 1 0 0 0 0
0 1 1 1 0 0 0
0 1 1 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 1 1 0 0
0 0 1 0 0 0 0
0 1 0 1 0 0 0
0 1 0 0 1 0 1
0 0 0 0 0 1 0
0 0 0 0 1 0 1
Success rate: 86.39

```

```
Enter the learning rate(between 0-1): 0.34
Enter the epoch number: 80
Threshold: 75
Press 1 for Perceptron rule
Press 2 for Delta Rule
2
Filenames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
1
All .txt files is opened from Binary_Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
Success rate: 100
```

```
Enter the learning rate(between 0-1): 0.6
Enter the epoch number: 90
Threshold: 70
Press 1 for Perceptron rule
Press 2 for Delta Rule
2
Filenames.txt is opened.
Press 1 for Binary input
Press 2 for Bipolar input
2
All .txt files is opened from Karakterler_Tam directory
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
1 0 0 0 0 0 0
0 1 0 0 0 0 0
0 0 1 0 0 0 0
0 0 0 1 0 0 0
0 0 0 0 1 0 0
0 0 0 0 0 1 0
0 0 0 0 0 0 1
Success rate: 100
```