T.C.

NAMIK KEMAL ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BMSB-434 PYTHON İLE DERİN ÖĞRENME

Yapay Sinir Ağını Radyan Cinsinden Sinüs Fonksiyonu Şeklinde Eğitmek

Emre PİLAVCI

Sertan KARAAĞAÇ

Ercan BULUŞ

TEKİRDAĞ-2020

|  |  |
| --- | --- |
| 1. **Yapay Sinir Ağını Radyan Cinsinden Sinüs Fonksiyonu Şeklinde Eğitmek…………………..** | 1 |
| 1. **Yapay Sinir Ağı Nedir?.......................................** | 2 |
| 1. **Yapay Sinir Ağının Özellikleri……...................** | 8 |
| 1. **Yapay Sinir Ağı Nasıl Eğitilir?...........................** | 12 |
| 1. **Uygulama……………………..............................** | 16 |
| 5.1 Relu Fonksiyonu Nedir?................................................ | 21 |
| 1. **Kaynaklar.............................................................** | 31 |

1. **Yapay Sinir Ağını Radyan Cinsinden Sinüs Fonksiyonu Şeklinde Eğitmek**

\*

1. **Yapay Sinir Ağı Nedir**

İnsan beyninin bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. Yapay Sinir Ağları ile basit biyolojik sinir sisteminin çalışma şekli taklit edilir. Taklit edilen sinir hücreleri nöronlar içerir ve bu nöronlar çeşitli şekillerde birbirilerine bağlanarak ağı oluştururlar.

Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiplerdir. Diğer bir ifadeyle, Yapay Sinir Ağları, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir.

Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiplerdir. Diğer bir ifadeyle, Yapay Sinir Ağları, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir.

Yapay Sinir Ağları, ağırlaştırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematiksel sistemlerdir. Transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır, bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır. Genelde işlem birimleri, kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar. Bu yapı da sinir ağlarını oluşturur.

Sinirsel hesaplamanın merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. Yapay Sinir Ağları, geleneksel işlemcilerden farklı şekilde işlem yaparlar. Geleneksel işlemcilerde, tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. Yapay Sinir Ağları ise her biri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimlerinden oluşmaktadır.

En basit şekilde, bir işlem birimi, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, işlem birimlerinin çalışma şekli yanıltıcı şekilde basittir. Sinirsel hesaplamanın gücü, toplam işlem yükünü paylaşan işlem birimlerinin birbirleri arasındaki yoğun bağlantı yapısından gelmektedir. Bu sistemlerde geri yayılım metoduyla daha sağlıklı öğrenme sağlanmaktadır.

Yapay Sinir Ağları’nın ana ögesi olan matematiksel fonksiyon, ağın mimarisi tarafından şekillendirilir. Daha açık bir şekilde ifade etmek gerekirse, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının şekli belirler.

Yapay Sinir Ağları’nın davranışları, yani girdiyi çıktıyla nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir.

1. **Yapay Sinir Ağının Özellikleri:**

Yapay sinir ağları aşağıdaki temel özelliklere sahiptir:

1. Doğrusal Olmama
2. Paralel Çalışma
3. Öğrenme
4. Genelleme
5. Hata Toleransı ve Esneklik
6. Eksik Verilerle Çalışma
7. Çok Sayıda Değişken Parametre ve Kullanma
8. Uyarlanabilirlik

Yapay Sinir Ağları uygulamaları en çok tahmin, sınıflandırma, veri ilişkilendirme, veri yorumlama ve veri filtreleme işlemlerinde kullanılmaktadır. Bunları tanımlayacak olursak;

Tahmin: Bu prensipte çalışan yapay sinir ağları girdi değerinden çıktıları tahmin etme üzerine çalışır, örneğin altın ons fiyatının tahmini.

Veri Filtreleme: Bu doğrultuda kodlanan yapay ağlar toplanan veriler arasından en işe yarayan verileri kullanır.

Sınıflandırma: Girdi değerlerini sınıflandırarak sistemin daha hızlı sonuca varmasına etkide bulunur.

Veri Yorumlama: Önceden eğitilen ağ girdilerini analiz eder, bir olay hakkında bu girdiler sayesinde yeni yorumlamalar yapabilmektedir.

Veri İlişkilendirme: Öğrendiği bilgilerle konuları ilişkilendirir ve bunun sonucunda ortaya çıkan eksik bilgileri tamamlar.

Yapay sinir ağlarının üstünlüklerinin yanı sıra bazı sakıncaları da vardır. Bu sakıncalar şu şekilde listelenebilir:

Sistem içerisinde ne olduğu bilinemez.

Bazı ağlar hariç kararlılık analizleri yapılamaz.

Farklı sistemlere uygulanması zor olabilir.

**4. Yapay Sinir Ağı Nasıl Eğitilir?**

Bir yapay sinir ağını oluşturan sinir hücreleri (nöronlar) sinapsisler aracılığı ile birbirlerine bağlıdırlar. Bu sinapsisler bağladıkları nöronların birbirinden haberdar olmasını sağlarlar. Ancak biyolojik sinir hücrelerinde olduğu gibi bilgisayar dünyasında bulunan yapay sinir ağlarında da bütün bağlantıların (sinapsislerin) ağırlığı eşit değildir. Bu ağırlık farklılıkları sinir ağının başarısını da etkileyen ve eğitim aşamasında belirlenen eğrilere sahiptir.

İşte yapay sinir ağının eğitimi konusu da tam bu noktada devreye girmektedir. Buna göre eğitim aşamasında bu değerlerin değiştirilmesi sonucu etkileyecek ve ağımız eğitilmiş olacaktır.

Bu eğitim işlemini, nöronların ve sinapsislerin değerlerini taşıyan verileri tutan matrislerin kullanılması ve matrisler üzerinde tanımlı olan matematiksel işlemlerin yapılması işlem hızı kazandıracaktır.

  Bir uzman tarafından sisteme elle girilen veriler eğitim süresini kısaltmaktadır. Bu tarz insan müdahalesi bulunan eğitim sistemlerine gözetimli (supervised) sistemler adı verilirken, yapay sinir ağının tamamen kendi başına bırakıldığı ve sonuç ağırlıklarına kendiliğinden ulaştığı sistemlere gözetimsiz (unsupervised) adı verilir.

Yapay sinir ağlarının fikir babası olarak bilinen, ‘Makineler düşünebilir mi?’ sorunsalını ortaya atan ünlü İngiliz matematikçi ve bilgisayar bilimci Alan Mathison Turing’dir. Aynı zamanda Turing Testi’ nin sahibidir.

Alan Turing’ i daha yakından tanımak isteyenler için «The Imitation Game» filmini izleyebilirsiniz.

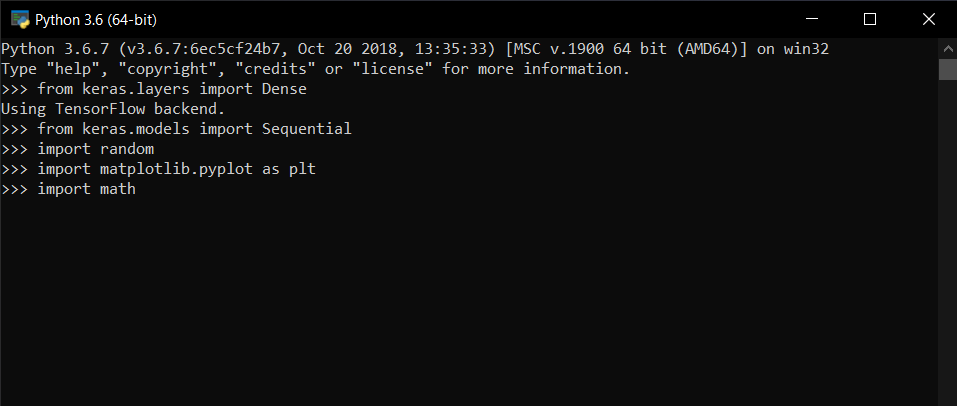
**5. Uygulama**

Python ile tensorflow & keras kullanarak basit bir şekilde yapay sinir ağımızı radyan cinsinden sinüs fonksiyonu görevi görecek şekilde eğiteceğiz.

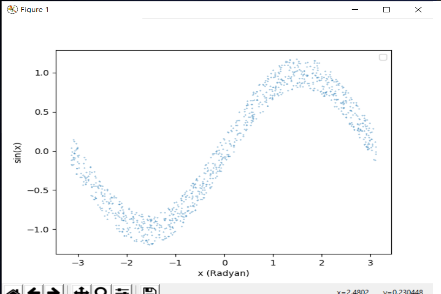
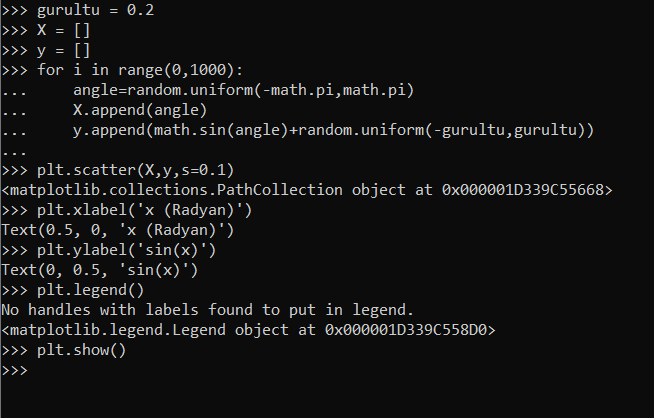
Radyan: Bir dairede yarıçap uzunluğundaki yay parçasını gören merkez açıya eşit açı ölçme birimidir. Açısal ölçünün standart birimidir ve matematiğin birçok alanında kullanılır. Bir açının radyan olarak ölçümü sayısal olarak bir birim dairenin karşılık gelen bir yayının uzunluğuna eşittir.

Sinüs: Merkezi orijin olan 1 birim yarıçaplı çember üzerindeki bir noktanın y eksenine göre koordinatıdır.

Öncelik olarak kullanacağımız kütüphaneleri sinir ağımızı eğitmek ve doğru çalıştırmak için projemize dahil ediyoruz.



Bu sinir ağımızı eğitmek için bir eğitim verisine ihtiyaç duymaktayız. Rastgele bin adet açı oluşturup oluşturulan açıların gerçek sinüs değerlerini math.sin fonksiyonu ile hesaplayarak yakınsak değerlerde bile tahminin güçlü olacağını kanıtlamak için 0.2’lik bir gürültü ekliyoruz ve oluşturulan dataseti çiziyoruz.

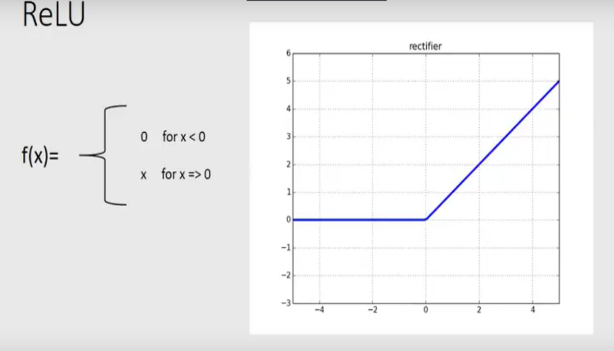


Sonrasında ise kullanacağımız modeli tanımlıyoruz. Modelin önceki adımında oluşturduğumuz datasetteki X açı değerlerini giriş olarak alacağız. Verilerimiz tek özelliğe sahip olduğu için bir giriş kullanacağız.

İlk katmanda 100 nöron olacak ve bu nöronların çıkışında relu aktivasyon fonksiyonu kullanılacaktır. Son katmanımızda ise tek bir nöron bulunacak ve bu nöronlar bizim y çıkış değerimizi verecektir.

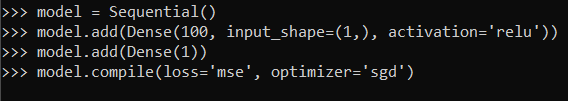
**5.1 Relu Fonksiyonu Nedir?**

Bu aktivasyon çok işlem gerektiriyor demektir. Ağdaki bazı nöronların aktif olup, aktivasyonun verimli bir hesaplama yükü olsun isteriz. ReLU ile bunu sağlamış oluruz. Negatif eksende 0 değerlerini alması ağın daha hızlı çalışacağı anlamına da gelmektedir.



Loss fonksiyonu olarak Mean Square Error ve optimizer olarak Stochastic Gradient Descent kullanılacaktır.

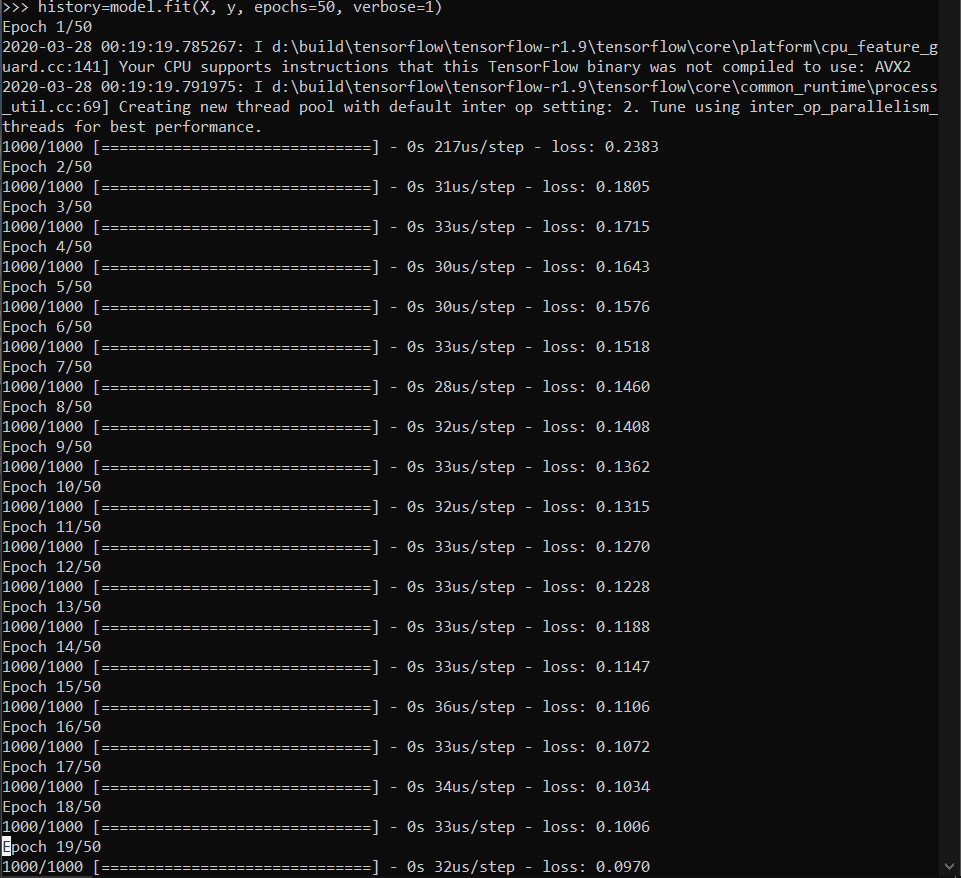
Loss fonksiyonu: Tasarlanan modelin hata oranını aynı zamanda başarımını ölçen fonksiyondur. Derin ağların son katmanı loss fonksiyonun tanımlandığı katmandır.



Mean Square Error: Yapay Sinir Ağları’nda modellerin performansını değerlendirmek için kullanılan bir indistir. burada ölçüm değerleri ile model tahminleri arasındaki hata oranını belirlemek için kullanılır. bu değerinin sıfır olması oluşturulan modelin mükemmel olması demektir.

Stochastic Gradient Descent: Destek vektör makineleri ve lojistik regresyon gibi konveks kayıp fonksiyonları altında lineer sınıflandırıcıların ayırt edici öğreniminde basit ama çok etkili bir yaklaşımdır.

Model eğitimi yanda görülen görseldeki gibi başlıyor ve eğitim 50 epoch şeklinde olacak ve verbose argümanıyla süreci izleyebileceğiz.

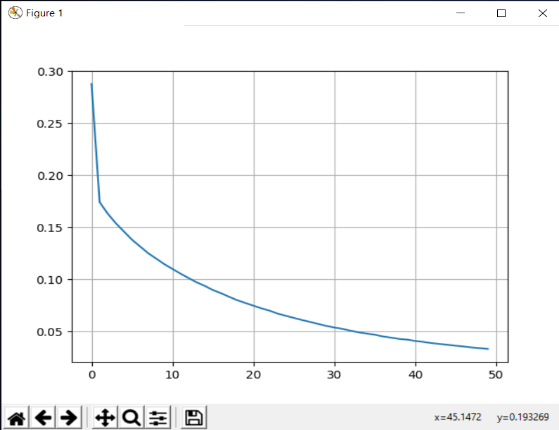
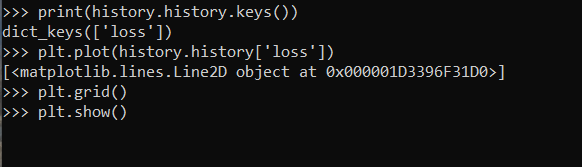


Verbose argümanı: Genel olarak ayrıntılı, programdaki bir işlem hakkında daha fazla bilgi raporlamak için kullanılabilecek isteğe bağlı bir argümandır.

Epoch (Eğitim Tur Sayısı): Model eğitilirken verilerin tamamı aynı anda eğitime katılmaz. Belli sayıda parçalar halinde eğitimde yer alırlar. İlk parça eğitilir, modelin başarımı test edilir, başarıma göre geriye yayılım ile ağırlıklar güncellenir. Daha sonra yeni eğitim kümesi ile model tekrar eğitilip ağırlıklar tekrar güncellenir. Bu işlem her bir eğitim adımında tekrarlanarak model için en uygun ağırlık değerleri hesaplanmaya çalışılır. Bu eğitim adımlarının her birine “epoch” denilmektedir.

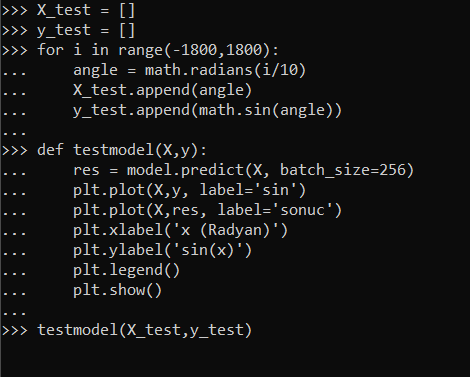
Şimdi ise loss değerinin değişimini çizdiriyoruz.

İyi tahmin eden bir model oluşturmamışsak, gerçek değer ile tahmin edilen değer arasındaki fark yüksek olacak dolayısıyla loss değeri de yüksek olacaktır.

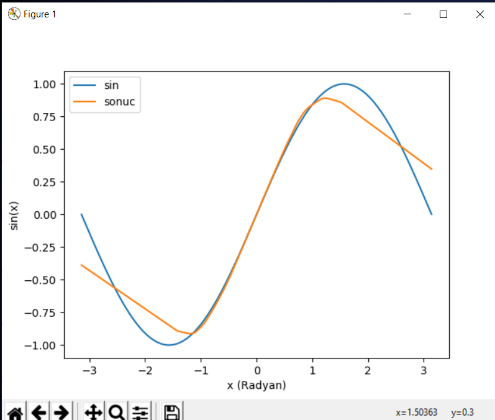


Model eğitimi bu aşamaya gelmeden tamamlandı. Sıra test aşamasına geldi.

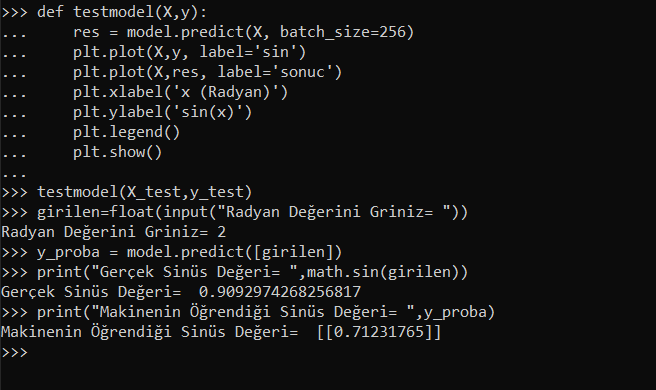
Bunun için gerçek değerlerden oluşan bir test verisi oluşturup testi başlatıyoruz.



Sonuç olarak gerçek sinüs fonksiyonu (mavi) ve ağın tahmin ettiği değerleri (turuncu) görmekteyiz.



Sinir ağı modelimizi kendi girdiğimiz bir değer ile test ettiğimizde gerçek sinüs değeri ile yapay sinir ağımızın öğrendiği sinüs değerini ekranda görüyoruz.



Sonuç olarak radyan değerini 2 girdiğimizde gerçek sinüs değeri yaklaşık 0.91 çıkarken makinemizin öğrendiği sinüs değeri yaklaşık 0.71 olarak karşımıza çıkmaktadır.

**6. KAYNAKLAR**

[**http://www.youtube.com**](http://www.youtube.com)[**http://.udemy.com**](http://.udemy.com)[**https://tr.wikipedia.org/**](https://tr.wikipedia.org/)[**https://www.linkedin.com/**](https://www.linkedin.com/)

[**https://www.cnet.com**](https://www.cnet.com/)