



Fenerbahçe Üniversitesi
BLM 102 – Algoritmalar ve Programlama II
Yapay Sinir Ağları ile Covid-19 Sağkalım Öngörüsü
Proje İçeriği

Veriliş Tarihi: 10.05.2020

Teslim Tarihi ve Yeri: 10.06.2020, Ders Saatlerinde, Elektronik olarak

1. Tanım:

Covid-19 salgını nedeniyle enfekte olup iyileşen ve hayatını kaybeden kişilerin çeşitli özellikleri kullanılarak, hastalıklı kişilerin hastalığı atlattığı ve atlayamayacağı (sağkalım) öngörüsünü yapan bir sistem geliştirilecektir. Bu analizin yapılması için Yapay Sinir Ağları (Neural Networks) algoritması gerçekleştirilecektir. Geliştirilen sistemin başarımı ve hastalık özellikleri kombinasyonları sisteme beslenerek özelliklere göre sağkalım analizleri yapılacaktır.

2. Proje Ekibi:

Proje 4 kişiye kadar olan ekiplerden oluşacaktır. Her bir proje ekibinin bir sorumlusu olacaktır. Öğrenciler 1, 2, 3 veya 4 kişilik kendi proje ekiplerini ve proje sorumlusunu belirlemelidirler.

Ekiplerin kurulması ve proje sorumlusunun belirlenmesi en geç **12.05.2020** tarihine kadar tamamlanmalıdır. Ekip sorumluları, LMS’te açılmış olan “Proje Ekip Sorumlularının Takımlarını Bildirmesi” başlığının altına, ekip üyelerinin isimlerini göndermelidirler.

LMS adresi: <http://levent.tc/lms/>

3. Kullanılacak Araçlar:

Proje kapsamında, ödev ve LAB’larda kullanılan Microsoft’un derleyicisi olan Visual Studio Community kullanılacaktır.

4. Tasarım Gereksinimleri

Covid-19 salgınında kişiler hastalığa yakalandıklarında sağkalım öngörü sistemi yapılacaktır. Bunun için nCov-2019 Veri çalışma grubunun (<https://github.com/beoutbreakprepared/nCoV2019>) sağladığı veriseti (https://github.com/beoutbreakprepared/nCoV2019/tree/master/latest_data) kullanılacaktır. Bu veriseti internet üzerinde açık olarak yayınlamış haberlerdeki hasta bilgileri toplanarak oluşturulmuştur. Verisetinde bazı boş ve projede kullanılmayacak alanlar temizlenerek aşağıda eğitim (training) ve test veriseti olarak verilmiştir.

http://levent.tc/courses/blm102/projeler/proje1/covid19_training_dataset.csv

http://levent.tc/courses/blm102/projeler/proje1/covid19_test_dataset.csv

Öğretim Elemanı: Dr. Vecdi Emre Levent, emre.levent@fbu.edu.tr, İzinsiz Kopyalanamaz

Eđitim veriseti 150, test veriseti ise 39 hasta bulunmaktadır.

Veriseti dosyasını indirdiđinizde, her bir satırda noktalı virgüller ile ayrılmıř 7 özelliđin olduđu görülecektir. Bu özellikler:

1. ID: Hastayı tanımlayan benzersiz bir numara
2. age: Hastanın yaşı
3. sex: Cinsiyet bilgisi
4. country: Hastanın ülkesi
5. chronic_disease_binary: Hastanın kronik rahatsızlıđının olup olmadıđı bilgisidir, false olmadıđı, true olduđu anlamına gelmektedir
6. outcome: Hastanın iyileřtiđi veya vefat ettiđi bilgisidir
7. source: Hastanın bilgisinin alındıđı bilgi kaynađı

Covid-19 sađkalım öngörüm sisteminin modelini oluřturmak için yapay sinir ađları algoritması kullanılacaktır. Yapay sinir ađlarına eđitim amacıyla eđitim verisetindeki 150 eđitim verisi kullanılarak model oluřturulacaktır.

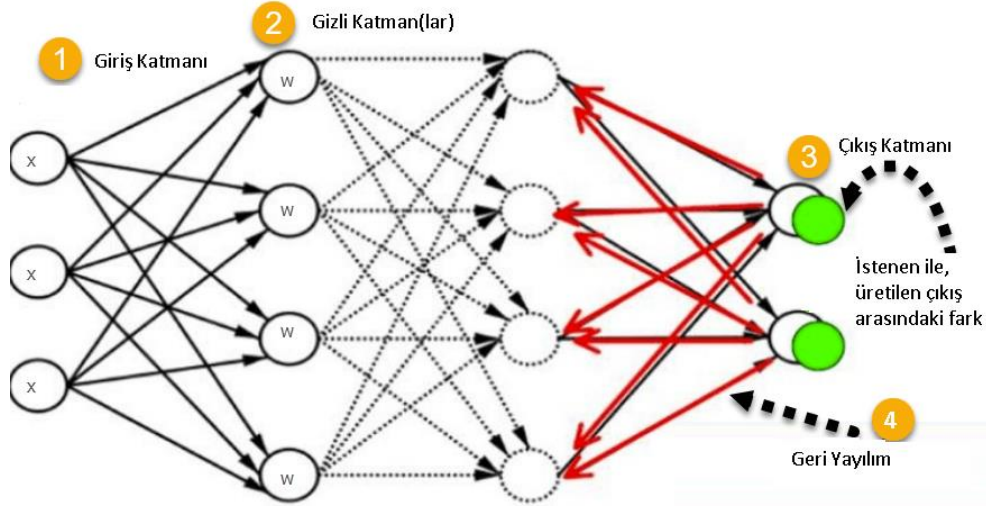
Test verisetinde ise 39 örnek vardır. Bu örnekler yapay sinir ađları ile oluřturulmuř modele beslenerek test edilecekler ve bařarım ölçümü yapılacaktır.

Yapay sinir ađları, birçok kompleks sistemin modellenebileceđi bir makine öđrenmesi algoritmasıdır. 1950'lerin bařında geliřtirilmeye bařlanmıřtır. Bu algoritma insan beynindeki sinir ađlarına benzemektedir. Bir sinir ađının çıktıısı, o sinir ađına giren veriler ve sinir ađı boyunca geçtiđi nöronların davranıřlarına göre deđiřmektedir. Nöron sayısı ve katman sayısı çok oldukça daha karmařık problemler çözüme yeteneđi artmaktadır. Ortalama bir insanda 100 milyar nöron varken, bir farede 75 milyon nöron olması, insan'ın daha karmařık problemleri çözebilme yeteneđini getirmektedir. Sadece nöron sayısı önemli deđildir. Nöronların dizilimleri oluřturduđu katmanlarda oldukça önemlidir. Kısacası, Yapay Sinir Ađları Nöron ve dizilimlerinin modellendiđi, istenilen giriř ve çıkıř sayılarının belirlendiđi beyinden esinlenerek geliřtirilmiř bir makine öđrenmesi algoritmasıdır.

Yapay sinir ađlarında öđrenme süreci, ilk katman olan giriř katmanından verilerin beslenmesi ile bařlar. Giriř katmanı sisteme beslenecek özelliklerin sayısı kadar nöron içermektedir.

Örneđin 10 adet kan parametresine bakılarak bir kanser tespiti ve kalan ömrün tahmin edileceđi bir sistem yapılacaktır. Sistemin giriři 10 adet özellik içermekte (kan deđerleri), 2 adet çıkıř (kanseri olup olmaması ve kalan ömür) vermektedir.

Yapay sinir ađına verilen giriřler, ara katmandaki nöronlar arasında transfer edilirler. Ara katmanlardaki nöron sayısı çok olması daha karmařık verisetlerinin modellenebilmesi için önemlidir. Eđer nöron sayısı az olursa, eđitim hiçbir zaman bařarılı olmayabilir. Ancak çok fazla olduđu zaman eđitim süresi çok uzayabilir. Dolayısıyla nöron-eđitim zamanı dengesi kurulmalıdır. řekil 1'de genel bir yapay sinir ađları modeli verilmektedir.



Şekil 1. Yapay Sinir Ağları Modeli

Şekil 1'deki örnek modelde 3 giriş ve 2 çıkış bulunmaktadır. Eğitim sürecinde eğitim veriseti içerisinde giriş verileri alınarak modele beslenir, modelin ürettiği çıkışlar ile, olması gereken çıkışlar karşılaştırılarak geriye yayılım (backpropagation) işlemi yapılır. Geriye yayılım işlemi, yapay sinir ağlarının içerisindeki nöronların katsayılarının güncellenmesi işlemidir. Eğitim tekrarlandıkça giriş ve çıkış arasındaki fark minimize edilmeye çalışılmaktadır.

5. Gerçekleme Yöntemi:

Bu proje kapsamında yapay sinir ağları kütüphanesi verilmiştir. Referans başlangıç kodunu aşağıdaki linkten indirebilirsiniz.

http://levent.tc/courses/blm102/projeler/proje1/YSA-COVID-19_baslangic_kodlari.rar

Verilen başlangıç kodunda yapay sinir ağları kütüphanesi projeye eklenmiş ve içerisindeki sınıflardan bazı objeler türetilmiştir.

```
uint32_t const numInputs = 16;
uint32_t const numHidden = 16;
uint32_t const numOutputs = 3;

Network::Settings networkSettings{ numInputs, numHidden, numOutputs };
Network nn(networkSettings);
```

Kod Parçacığı 1

Kod parçacığı 1'de yapay sinir ağlarının katman nöron sayıları ayarlanmaktadır. Giriş katmanında verisetindeki özellik sayısı (numInputs) yazılmalıdır. Verilen kütüphane tek bir gizli katman desteklemektedir. Burada istenen sayıda nöron kullanılabilir. Buradaki nöron sayısını, eğitim işlemi yaptırdıktan sonra, eğitimin yeterince başarılı performans vermediği durumda artırılabilir. Çıkış nöron sayısı (numOutput) ise istenen çıkış sınıfını belirtmektedir. Network nn satırında, yapay sinir ağları oluşturulmuştur.

```
NetworkTrainer::Settings trainerSettings;
trainerSettings.m_learningRate = 0.001;
trainerSettings.m_momentum = 0.9;

NetworkTrainer trainer(trainerSettings, &nn);
```

Kod Parçacığı 2

Kod parçacığı 2’de yapay sinir ağıları parametrelerinden olan eğitim oranı ve momentum sayılarının ayaları yapılmaktadır. Bu sayılar eğitimin performansını etkilemektedir. Eğitim başarısının düşük olması durumunda değiştirilerek sonuçları gözlemlenebilir.

Kod parçacığı 2’den sonra yapay sinir ağlarının eğitim setine göre öğrenme işleminin yapılması için bir döngü oluşturulacaktır. Bu döngünün sonlanması için iki koşul olmalıdır. Bu koşullar maksimum döngü sayısı ve hedef başarıya ulaşmaması olmalıdır.

Yukarıda bahsedilen koşullara göre dönecek olan döngünün içerisinde hatalı tespit edilen çıktıların sayısını tespit etmek için double türünde bir değişken tanımlanıp 0’a eşitlenmelidir. Sonrasında eğitim verisetinin içerisinde tüm veriler satır satır okunarak “trainer.Train” fonksiyonuna beslenmelidir. Besleme işlemi için Train fonksiyonunun ilk argümanı olan “struct TrainingEntry” türünde verinin hazırlanması gerekmektedir. Bunun için kod parçacığı 3 kullanılabilir.

```
struct TrainingEntry currData;  
  
currData.m_inputs = //Double türünde giriş vektörü  
currData.m_expectedOutputs = // Int türünde çıkış vektörü
```

Kod parçacığı 3

Struct TrainingEntry türünde bir obje türetilmiştir. Bu objenin m_inputs bölümüne, eğitim dosyasından okunmuş ve double vektörü haline getirilmiş vektör atanır. m_expectedOutputs bölümüne ise eğitim dosyasından okunmuş olan satırdaki çıkış olarak beklenen integer vektörü verilir. Train fonksiyonunun ikinci argümanı olan MSE(Mean Square Error) için bir double değişken açılıp verilir. Eğitim işleminin sözde kodu (Pseudocode) Kod parçacığı 4’te verilmiştir.

```
while(dongu < maxDongu && dogruluk<hedefDogruluk){  
  
    yanlisTespitler = 0;  
  
    MSE = 0;  
  
    for(...) {  
        //Her bir döngüde dosyadan bir satır okuyup, giriş ve çıkış vektörü hazırlanmalıdır.  
  
        struct TrainingEntry currData;  
        currData.m_inputs = // double türündeki giriş vektörü  
        currData.m_expectedOutputs = // int türündeki çıkış vektörü  
        trainer.Train(currData, MSE, incorrectEntries);  
  
    }  
  
    dogruluk = 100.0 - (yanlisTespitler / toplamOrnekSayisi * 100.0);  
    dongu++;  
  
    cout << “Dongu: “ << dongu << “Dogruluk :” << dogruluk << “ “ << MSE << “\n”;  
  
}
```

Kod Parçacığı 4

Eğitim maksimum döngü veya eğitim başarısına ulaşarak sonlanmasından sonra, test işlemi yapılmalıdır. Eğitimde olduğu gibi test dosyasından veriler satır satır okunarak “nn.Evaluate” fonksiyonu çağrılmalıdır. Evaluate fonksiyonu sadece double türündeki giriş vektörü almaktadır. Yani dosyadan okunduktan sonra sadece giriş için kullanılacak özelliklerden oluşturulmuş bir vektör beslenecektir. Evaluate fonksiyonu geriye vektör döndürmektedir. Bu vektör tespit edilen çıkış vektörüdür. Olması gereken ile evaluate fonksiyonunun çıktısını karşılaştırarak tüm test verisetinin

doğruluk yüzdesi ekrana gösteriniz. Gösterilen yüzde, hasta bilgileri ile sağkalım tespitinin ne kadar başarılı yapılabildiğini gösterecektir.

Verilen verisetinde eğitime sokulacak giriş özellikleri:

- Yaş
- Cinsiyet
- Kronik rahatsızlık

verileridir.

Çıkış olarak tek bir özellik olan outcome verisi kullanılacaktır. Yapay sinir ağlarına beslenecek verilerin numerik olması gerekmektedir. Bunun için cinsiyet değerinde male okunduğunda 0, female okunduğunda 1 değeri; kronik rahatsızlıkta false okunduğunda 0, true okunduğunda 1 verilmelidir. Çıkış için source bölümünde ise discharged okunduğunda 0, died okunduğunda ise 1 verilmelidir.

Tüm verilerin 0-1 arasında normalize edilmesi gerekmektedir. Yaş verisi haricindeki veriler 0-1 arasındadır. Ancak yaş verisinin normalize edilmesi gerekmektedir. Bunun için dosyadan tüm verilerin okunup, en küçük ve en büyük kişilerin yaşları bulunup, en küçük kişinin yaşının 0'a, en büyük kişinin yaşının 1'e diğerlerinin ise 0-1 arasında olacak şekilde bir matematiksel formül ile hesaplanmalıdır.

Tasarım yapılırken nesneye yönelimli programlama (sınıflar, kalıtım, çok biçimlilik) ve modüler programlama (operatör aşırı yükleme, şablonlar, istisna idaresi) yaklaşımlarını mümkün oldukça kullanınız. Proje teslim raporunuzda geliştirilen tasarımın mimarisini gösteren UML diyagramını veriniz.

Not: Tasarım test edilirken, proje Visual Studio'da geliştirilirken Debug modunda çalıştırıldığında yavaş çalışabilir, hızlandırmak için release x64 modunda çalıştırılabilir.

6. Notlandırma ve Proje Teslimi:

Bu başlık "Yapay Sinir Ağları ile Covid-19 Sağkalım Öngörüsü" projesinin teslimi ve notlandırılması hakkında bilgiler içermektedir.

6.1. Notlandırma:

Projenin iki ana değerlendirme kriteri vardır. Her iki kriter 50 şer puandır.

İlk kriter "Yapay Sinir Ağları ile Covid-19 Sağkalım Öngörüsü" 'nin **doğru çalıştırılması ve objeye dayalı programlama dili yaklaşımlarının kullanılmış olmasıdır.**

İkinci kriter ise **Proje Teslim Dokümanı ve Sunumdur.**

- **Proje Teslim Dokümanı:**

Öğrenciler, proje raporlarını verilen "Proje Teslim Dokümanı" 'nın içerisini doldurarak yapacaklardır.

Proje Teslim Dokümanı: http://levent.tc/courses/blm102/BLM102_proje_teslim_dokumani.doc

Proje teslim dokümanı tamamlanıp PDF'e dönüştürüldükten sonra, PDF'in sonuna Latex (Genellikle bilimsel çalışmaların yazımında kullanılan, görsel olarak Word ile yazılmış dokümanlara göre çok daha başarılı olan bir doküman oluşturma dilidir) ile tasarlanmış özgeçmişler eklenecektir. Özgeçmişin minimumda kullanılan diller, yabancı diller, projeler (kısa açıklamaları ile), kullanılan kütüphaneler (varsa, örn. OpenCV), not ortalaması, github adresi, youtube adresi bilgileri verilmelidir.

Öğretim Elemanı: Dr. Vecdi Emre Levent, emre.levent@fbu.edu.tr, İzinsiz Kopyalanamaz

Latex ile özgeçmiş oluşturma: http://levent.tc/courses/blm102/BLM102_latex_ozgecmis_olusturma.pdf

Latex ile özgeçmiş oluşturulduğunda PDF uzantılı dosyalar elde edilecektir. Proje teslim dokümanının PDF'inin arkasına eklemek için PDF birleştirici araçları (https://www.ilovepdf.com/merge_pdf gibi) kullanılmalıdır. İşlem sonucunda tek bir PDF dosyası elde edilecektir.

- **Proje Sunumu:**

Powerpoint üzerinde ortalama 5 dakika (4-6 dakika arası) sürecek bir sunum hazırlayarak kayıt etmelidirler. Kayıt işlemi, cep telefonu veya bilgisayar ekran kayıt yazılımları (Screen-Recorder, Bandicam vb...) ile yapılabilir.

Sunum, ekip üyeleri içinden biri tarafından, projenin nasıl yapıldığı, nasıl çalıştığı vb.. konularının powerpoint slaytları üzerinden anlatılırken kaydedilmesi ile olmalıdır. Sunum video'sunda powerpoint slaytları okunabilir ve konuşmacının sesinin anlaşılır olması gerekmektedir. Powerpoint slayt görünüm tasarımı istenildiği gibi yapılabilir.

Proje ekibinin tamamı, notlarını bu değerlendirmeye göre alırlar.

6.2. Teslim:

Projenin teslimi için aşağıdaki adımların gerçekleştirilmesi gerekmektedir. İstenen dosyaları sadece proje ekip sorumlusunun getirmesi, LMS ve Github (Çok yaygın bir açık kaynak kod paylaşım platformudur)'a yüklemelidir.

Proje Teslim Dokümanının, ders sınıfı ve saatinde, çıktılarının alınarak teslim edilmesi gerekmektedir.

Ayrıca LMS'te açılmış olan "Proje Teslim" sayfasına aşağıdaki dosyaların yüklenmesi gerekmektedir.

- Proje kaynak kodları
- Hazırlanan powerpoint sunum dosyası (.ppt uzantılı dosya)
- Proje Teslim Dokümanı (Word formatında yüklenmelidir)
 - Dokümanın alt başlıkları doldurulmalıdır. (Örnek proje dokümanı: http://levent.tc/courses/blm103/projeler/2019-2020-tamamlananlar/BLM103_yapay_zeka_rapor_ozlem_deniz.pdf)
 - Kaydedilen powerpoint sunum video'su youtube'a yüklenip, adresi, dokümanın sonuçlar bölümündeki açılmış yere link'i yazılmalıdır (Video'nun herkes'e görünür olmamasını istiyorsanız, youtube'a yükledikten sonra liste dışı seçeneğini seçerek, sadece link'e sahip olan kişilerin görmesini sağlayabilirsiniz).
 - LMS'e yüklenen tüm dosyalar (Proje kaynak kodları, ppt uzantılı sunum dosyası ve Proje Teslim Dokümanını (PDF formatında)), github.com sitesine üye olup, yüklenip, dokümanın sonuçlar bölümündeki yere link'i yazılmalıdır.