



ÇORLU MÜHENDİSLİK FAKÜLTESİ
BİYOMEDİKAL MÜHENDİSLİĞİ BÖLÜMÜ

PARKİNSON TANI KALEMİ

Merve TEMUR

LİSANS BİTİRME ÇALIŞMASI

2020



ÇORLU MÜHENDİSLİK FAKÜLTESİ
BİYOMEDİKAL MÜHENDİSLİĞİ BÖLÜMÜ

PARKİNSON TANI KALEMİ

Merve TEMUR

Biyomedikal Mühendisliği
ünvanı için
Biyomedikal Mühendisliği Bölümüne
teslim edilmiştir.

2020

PARKİNSON TANI KALEMİ

ONAYLAYANLAR:

Doç. Dr. Koray Rifat ÇİFTÇİ
(Tez Danışmanı)

Yrd. Doç. Dr. İsmail DEVECİOĞLU
.....

ONAY TARİHİ: 20 Temmuz 2020

TEŞEKKÜRLER

Lisans bitirme projesi tezinde ve de projenin prototipleme aşamalarında emeği geçenlere, yol gösterici olan Sayın Doç. Dr. Rifat Koray ÇİFTÇİ hocama teşekkür ederim.

ÖZET

PARKINSON TANI KALEMİ

Parkinson hastalığı, beyinde dopamin adı verilen, beyin hücrelerinin birbirleriyle haberleşmesini sağlayan maddeyi üreten hücrelerin bozulması sonucu ortaya çıkar. Beyinde dopamini üreten hücreler hareketlerin kontrolünden, uyumundan ve akıcılığında sorumludur. Hareketlerde yavaşlık, dinlenme halindeyken titreme, psikiyatrik rahatsızlıklarla kendini belli eder. Bu projede kaleme aç-ıvme sensörü takılarak tanı yapılacak kişinin el titreme şiddeti tespit edilmeye çalışılmıştır. Titreme şiddetinden yola çıkarak hastalığın tanısına ulaşmak amaçlanmıştır. Tanı konacak kişinin bu kalemle belirli referansları olan çizimler yapılması istenmiştir. Çizim sırasında sensör ivme değerlerini kaydedilerek yapay sinir ağına girişi yapılır. Sensörün ölçtüğü ivme değerlerinden yola çıkarak ağın tanı çıktısı vermesine çalışılmıştır. .

Anahtar Sözcükler: Parkinson, Aç-ı, İvme, Sinir Ağı

ABSTRACT

PARKINSON DIAGNOSTIC PEN

Parkinson disease is caused by the breakdown of cells that produce the substance called dopamine in the brain that allows brain cells to communicate with each other. Cells that produce dopamine in the brain are responsible for the control, adaptation and fluency of movements. In this project, an angle-acceleration sensor was attached to the pen and the hand tremor intensity caused by the recognition was tried to be determined. It is aimed to choose the diagnosis of the disease based on the tremor intensity. The person to be diagnosed was asked to make drawings with this reference. Sensor acceleration values are recorded during drawing Based on the acceleration values measured by the sensor, the network was tried to give a diagnostic output.

Keywords: Parkinson, Angle, Acceleration, Neural Network

İÇİNDEKİLER

| | |
|--|------|
| TEŞEKKÜRLER | iii |
| ÖZET | iv |
| ABSTRACT | v |
| ŞEKİL LİSTESİ | vii |
| TABLO LİSTESİ | viii |
| 1 GİRİŞ | 1 |
| 2 TEORİ | 2 |
| 2.1 Parkinson Hastalığı Titreme | 2 |
| 2.2 İvmeölçer ve Jiroskop Sensörü | 2 |
| 3 METOTLAR | 4 |
| 3.1 AÇI VE İVME ÖLÇÜMLERİ | 6 |
| 3.1.1 Açık ve İvme Değerlerinin MATLAB Ortamına Ak- tarılması | 7 |
| 3.1.2 Verilerin Kaydedilmesi | 8 |
| 3.2 YAPAY SİNİR AĞLARI | 8 |
| 3.2.1 MATLAB Yapay Sinir Ağları | 8 |
| 3.2.2 Yapay Sinir Ağlarının Eğitilmesi | 9 |
| 3.2.3 Yapay Sinir Ağlarının Simüle Edilmesi | 12 |
| 4 SONUÇLAR ve TARTIŞMA | 15 |
| A MATLAB KODLARI | 17 |
| KAYNAKÇA | 22 |

ŞEKİL LİSTESİ

| | | |
|------|--|----|
| 2.1 | Açısal ivme ölçer | 3 |
| 2.2 | Gyro sensör | 3 |
| 3.1 | Referans şekil | 5 |
| 3.2 | 3 Boyutlu kalem tasarımı | 5 |
| 3.3 | Prototip ürün | 5 |
| 3.4 | Kalem ölçüme hazır | 6 |
| 3.5 | Kalem ölçüm sırasında | 6 |
| 3.6 | MPU-Arduino UNO bağlantı şeması | 7 |
| 3.7 | Arduino IDE seri portalda çizdirilen açı-ivme değerleri . | 7 |
| 3.8 | 20 Nörona sahip sinir ağının eğitim performans grafiği . | 9 |
| 3.9 | 20 Nörona sahip sinir ağının eğitim regresyon grafikleri . | 10 |
| 3.10 | 10 Nörona sahip sinir ağının eğitim performans grafiği . | 10 |
| 3.11 | 10 Nörona sahip sinir ağının eğitim regresyon grafikleri . | 11 |
| 3.12 | 20-20 Nörona sahip sinir ağının eğitim performans grafiği | 11 |
| 3.13 | 20-20 Nörona sahip sinir ağının eğitim regresyon grafikleri | 12 |
| 3.14 | Tek katmanlı 20 nörona sahip ağın Simülasyon çıktıları . | 12 |
| 3.15 | Tek katmanlı 30 nörona sahip ağın Simülasyon çıktıları . | 13 |
| 3.16 | Tek katmanlı 50 nörona sahip ağın Simülasyon çıktıları . | 13 |
| 3.17 | Çift katmanlı 20-20 nörona sahip ağın Simülasyon çık- tıları | 13 |
| 3.18 | Çift katmanlı 50-20 nörona sahip ağın Simülasyon çıktıları | 13 |
| 3.19 | BFGS quasi-Newton backpropagation öğrenme fonksi- yonu ile eğitilmesi sonucu simüle edilen ağın validasyon çıktıları | 14 |
| 3.20 | One step secant backpropagation fonksiyonu ile eğitil- mesi sonucu simüle edilen ağın validasyon çıktıları . . . | 14 |

TABLO LİSTESİ

Bölüm 1

GİRİŞ

Parkinson hastalığı, beyin hücrelerinin birbirleriyle etkileşime geçmesini sağlayan dopamin adlı kimyasal habercinin yeteri kadar üretilmemesi sonucu ortaya çıkar. Dopamin eksikliği Parkinson hastalığının belirtileri olan hareketlerde yavaşlama, dinlenme halindeyken fark edilen titremeler olarak kendini gösterir. Bu projede Parkinson hastalığının yaygın belirtilerinden olan titremenin şiddetinden yola çıkarak hastalığa dair ön tanıya ulaşmak amaçlanmıştır. El yazı muayeneleri parkinson hastalığında titreme analizinde yaygın olarak kullanılmaktadır. Bu bilgiden yola çıkarak bir kaleme açı-ivme sensörü entegre ederek titreme şiddetine ulaşılabilmektedir. Titreme şiddeti bize ön tanı imkanı sağlamaktadır.

Bölüm 2

TEORİ

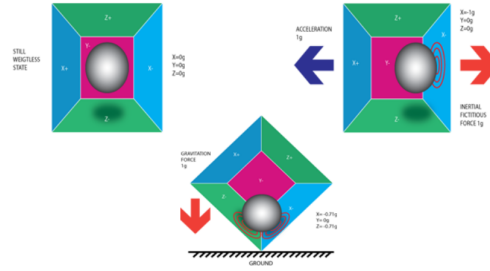
Bu kısım tezinizin alt yapısını oluşturan literatür bilgilerini içerir.

2.1 Parkinson Hastalığı Titreme

Dopamin üreten hücrelerin yüzde 60-80 oranında tahrip olmasıyla dopamin miktarı zamanla giderek azalır ve hastada hareketler giderek zorlaşmaya başlar.Hastalık ilk olarak bir elin özellikle ince hareketlerde beceriksizleşmesi, yavaşlaması ve dinlenme anında belirgin titreme şeklinde kendini gösterir.El yazı muayeneleri parkinson hastalığında titreme analizinde yaygın olarak kullanılmaktadır.

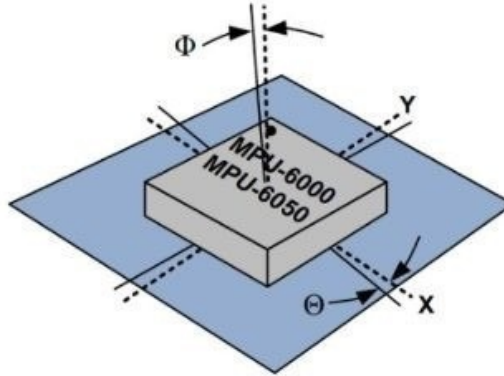
2.2 İvmeölçer ve Jiroskop Sensörü

MPU6050 3 eksenli gyro ve 3 eksen açısal ivme ölçer olan IMU sensörüdür.İvme ölçer Piezoelektrik etkisi ile çalışmaktadır. Piezoelektrik bazı malzemelerin üzerine mekanik olarak bir kuvvet uygulanması sonucunda ortaya çıkan elektriktir. Sensör kuvvet olarak yer çekimini kullanarak değer verir.Teknolojik cihazlarda titreşimi ölçer ve belirli aralıkta kalmasını sağlar.



Şekil 2.1: Açısal ivmeölçer

Jiroskop sensörü ise açısal hızı algılayabilen bir sistemdir. Yani sabit duran bir cismin, üç dikey ekseninde açısal oranlar karşılaştırılarak dönüş yönü ve hızı belirlenir. Algıladığı verileri işlemci sayesinde işleyerek elektriksel sinyale çevirir. Jiroskop, yön ölçümü veya ayarlamasında kullanılan, açısal dengenin korunması ilkesiyle çalışır. Jiroskopik hareketin temeli fizik kurallarına ve merkezkaç ilkesine dayalıdır.

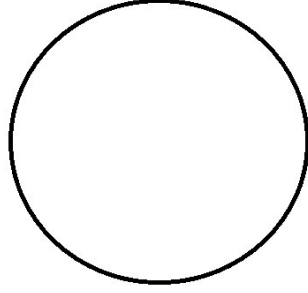


Şekil 2.2: Jiroskop

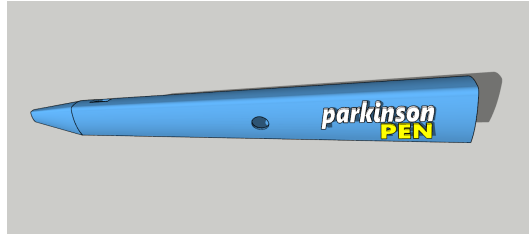
Bölüm 3

METOTLAR

Titreme şiddetini ölçmek için aç-ivme sensöründen yararlanılmıştır. Aç-ivme sensörü bir kaleme entegre edilerek tanı konulacak kişiden Şekil3.1.de bulunan referans şeklin üzerinden geçmesi istenir. Çizim esnasında sensör ivme ölçümü yapar. Elde edilen ölçüm değeri kaydedilir. Ön tanıda bulunacak yapay sinir ağı, sağlıklı bireylere ait ölçüm değerleri ve hastalıklı bireylere ait ölçüm değerleri kullanılarak eğitilir. Tanı esnasında toplanan ölçüm değerleri eğitilmiş yapay sinir ağına girilir ve yapay sinir ağının hasta veya sağlıklı olarak tanı çıktısı vermesi beklenir. Kalem tasarımı, ölçümün istenilen zamanda başlamasını kontrol etmek için üzerine entegre edilecek buton ve de ölçümde olup olmadığına dair bilgi vermesi için de ölçümde iken yeşil ışık ölçümde değilken kırmızı ışık saçan rgb led kullanılması göz önüne alınarak yapılmıştır. Kalem tasarımı 3 boyutlu modelleme programı olan Sketchup programı kullanılarak yapıldı. Kalem buton, rgb led ve MPU6050 içerisine yerleştirilecek ölçüde tasarlandı, tasarım Şekil 3.2’de görülmektedir. 3 boyutlu yazıcı kullanılarak model yazdırıldı ve elemanlar içerisine monte edildi.



Şekil 3.1: Referans şekil



Şekil 3.2: 3 Boyutlu kalem tasarımı



Şekil 3.3: Prototip ürün



Şekil 3.4: Kalem ölçüme hazır durumdayken



Şekil 3.5: Kalem ölçüm sırasında

3.1 AÇI VE İVME ÖLÇÜMLERİ

MPU-6050 açı-ivme sensörünün VCC pini Arduino UNO'nun 5V pinine, GND pini- GND pinine, SCD pini -Analog 5 pinine, SDA pini- Analog 4 pinine ve INT pini- Dijital 2 pinine denk gelecek biçimde Şekil 3.6' da gösterildiği devre bağlantısı yapıldı.

3.1.2 Verilerin Kaydedilmesi

Açı-ivme sensöründen elde edilen veriler matlab üzerinden text dosyası oluşturularak kayıt edildi. Ölçüm verilerini kaydetmek için kullanılan MATLAB kodu Ek 3'de bulunmaktadır.

3.2 YAPAY SİNİR AĞLARI

Yapay sinir ağları (YSA), insan beyninin bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit biyolojik sinir sisteminin çalışma şekli taklit edilir. Yani biyolojik nöron hücrelerinin ve bu hücrelerin birbirleri ile arasında kurduğu sinaptik bağın dijital olarak modellenmesidir. Nöronlar çeşitli şekillerde birbirlerine bağlanarak ağlar oluştururlar. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

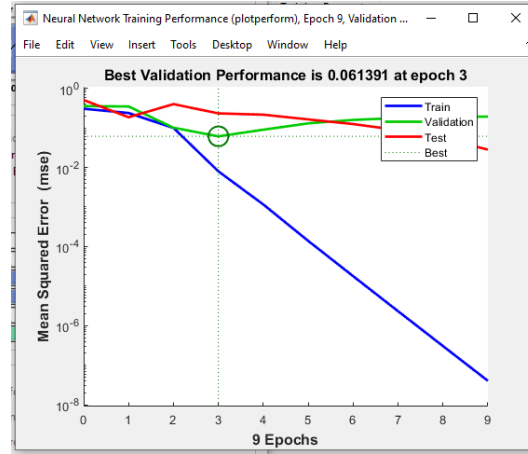
3.2.1 MATLAB Yapay Sinir Ağları

Grafiksel Arayüz MATLAB içerisinde kullanıcılar için oluşturulmuş kullanımı basit bir araçtır. Grafiksel Arayüz ile birlikte yapay sinir ağları için model kurulması, eğitilmesi ve test edilmesi mümkündür. Grafiksel Arayüz kullanımı için yapılması gereken MATLAB komut sayfasında "nntool" komutunu girmektir. Yapay sinir ağlarının hastalık tanısı yapabilmesi için sağlıklı bireylerin ölçüm değerleri ile hasta bireylerin ölçüm değerleri arasındaki farkı öğrenmiş olması yani ağın eğitilmesi gerekir. Ağ eğitme işlemleri veri setini belirleme, ağ yapısını oluşturma ve ağı eğitime sokarak belirli sonuçlar elde etme yöntemle-

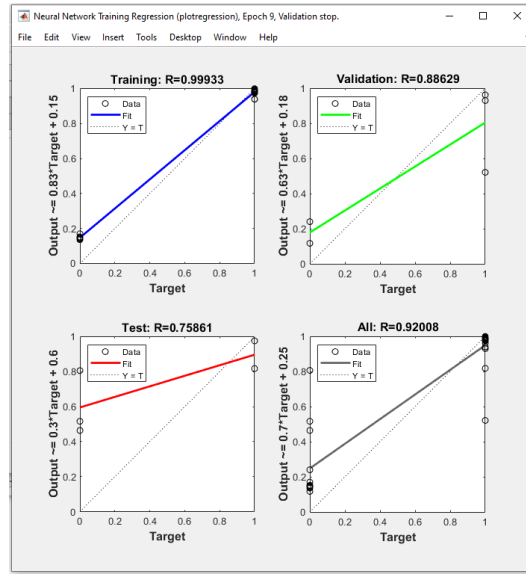
rinden oluşmaktadır.

3.2.2 Yapay Sinir Ağlarının Eğitilmesi

Yapay sinir ağı eğitimi için ilk olarak giriş ve çıkış veri setinin tanımlanması gerekir. İkinci olarak ise ağı yapısı yani katman sayısı ve bu katmanlarda bulunacak nöron sayıları tanımlanmalıdır. Ağı eğitimi için kullanılacak algoritma, iterasyon sayısı, performans değerini ve öğrenme oranı da tanımlanarak ağı eğitime hazır hale getirilir. Train komutuna gerekli parametreler yazılarak ağı eğitimi yapılır. İlk olarak yapay sinir ağına açıcı ivme sensöründen alınmış 30 adet sağlıklı bireylere ait ölçüm veri seti (100 ivme değeri) ve 20 adet hastalıklı bireylere ait ölçüm veri seti girildi. Veriler sütun matrisler haline getirildi. Ağı yapısı tek katman olacak şekilde 20 nöronlu oluşturuldu. Ağı eğitilmesinde kullanılacak öğrenme fonksiyonu 'trainlm', öğrenme oranı 0.8, döngü parametresi 1000, hedef $1e-25$ olarak tanımlandı. MATLAB kodları Ek 4'dedir. Eğitim işlemi sonucu elde edilen performans grafikleri Şekil 3.8-3.9'da görülmektedir.

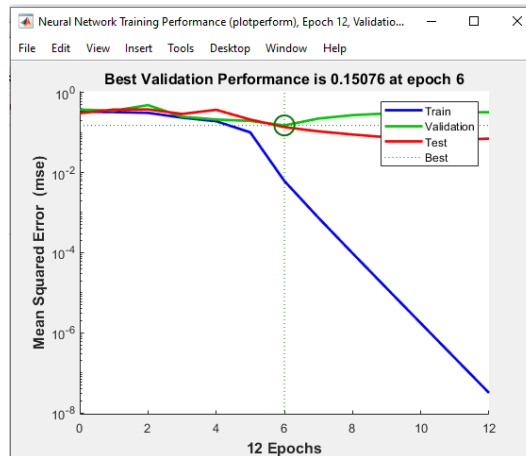


Şekil 3.8: 20 Nörona sahip sinir ağı eğitimi performans grafiği

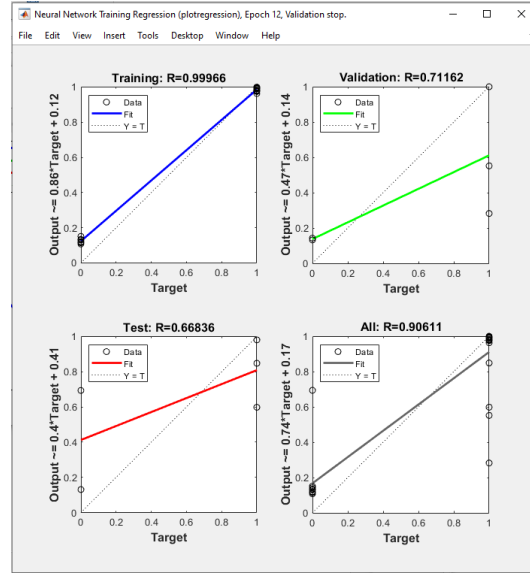


Şekil 3.9: 20 Nörona sahip sinir ağının eğitim regresyon grafikleri

Aynı veri seti yine tek katmanlı bir ağ yapısında ilk olarak 30 nöron daha sonra 50 nöron kullanarak ve diğer eğitim parametreleri sabit tutularak eğitildi. Ağın nöron sayısının artmasının, ağın öğrenme oranını arttırdığı tespit edildi. Sensörden alınan ivme değer sayısı 25'e düşürüldü. Ağ yapısı tek katmanlı ve 10 nöron olarak tanımlandı ve diğer eğitim parametreleri sabit tutularak ağ eğitildi. Eğitim sonucu elde edilen performans grafikleri Şekil 3.10-3.11'de görülmektedir.

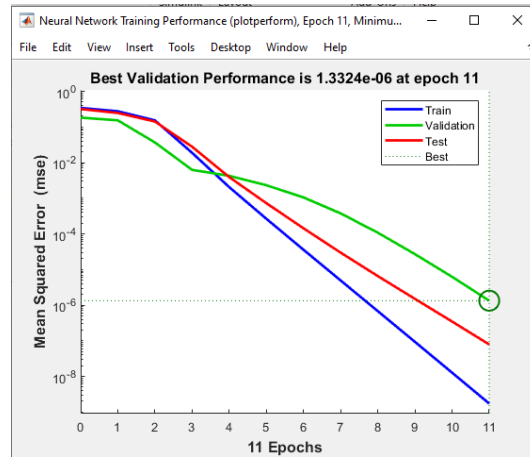


Şekil 3.10: 10 Nörona sahip sinir ağının eğitim performans grafiği

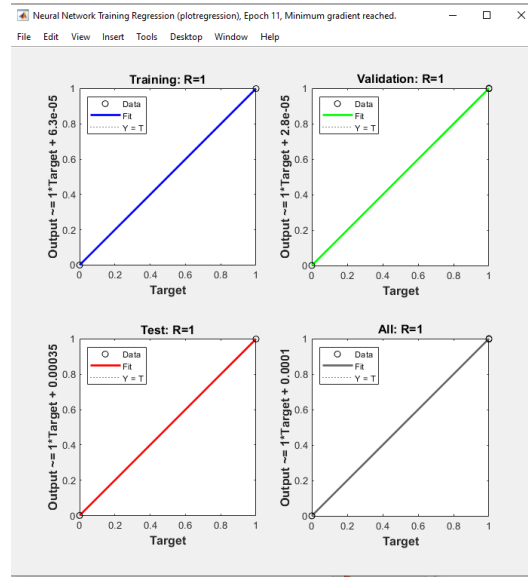


Şekil 3.11: 10 Nörona sahip sinir ağının eğitim regresyon grafikleri

İkinci veri seti, ağıın nöron sayısı sırasıyla 20, 30, 50 nörona çıkarılarak tekrar eğitildi. İlk veri setine oranla ikinci veri setinin öğrenme oranı nispeten düşük olsada yüksek öğrenme değerleri verdiği tespit edildi. Sinir ağında bulunan nöron katman sayısı 2'ye çıkarıldı ve ikinci veri seti diğer eğitim parametreleri sabit tutularak eğitildi. Eğitim sonucu elde edilen performans grafikleri Şekil 3.12-3.13'de görülmektedir.



Şekil 3.12: 20-20 Nörona sahip sinir ağının eğitim performans grafiği



Şekil 3.13: 20-20 Nörona sahip sinir ağının eğitim regresyon grafikleri

Çift katmanlı oluşturulan ağın nöron sayıları 20-20, 30-20, 50-20, 50-30 belirlenerek ağ eğitildi. Kullanılan veri seti için ideal nöron sayısı 20-20 olarak tespit edildi. İdeal nöron sayısı belirlendikten sonra ideal öğrenme fonksiyonunu tespit edebilmek için ağ BFGS quasi-Newton backpropagation, one step secant backpropagation ve Levenberg-Marquardt backpropagation fonksiyonları ile eğitildi.

3.2.3 Yapay Sinir Ağlarının Simüle Edilmesi

Eğitilmiş yapay sinir ağı test veri seti ile simüle edilerek ağın öğrenmesi test edilir. Şekil 3.14-20’de farklı nöron sayılarına ait Levenberg-Marquardt backpropagation öğrenme fonksiyonutla eğitilmiş ağların simülasyon çıktısı görülmektedir.

```
>> [vcikti,vld_hedef] = siniragi2(nl,loran, training_oran)

vcikti =

    1     0     0     1     0     0     0     0     1     0     0     0     1     1     0     1     1     0     0     0

vld_hedef =

    1     1     0     1     0     1     0     0     1     0     0     0     1     1     1     1     1     1     1     0
```

Şekil 3.14: Tek katmanlı 20 nörona sahip ağın Simülasyon çıktıları

```
>> [vcikti,vld_hedef] = siniragi2(n1,loran, training_oran)

vcikti =

    1    1    0    1    0    1    0    0    1    0    0    0    1    0    1    1    1    0    1    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.15: Tek katmanlı 30 nörona sahip ağın Simülasyon çıktıları

```
>> [vcikti,vld_hedef] = siniragi2(n1,loran, training_oran)

vcikti =

    1    1    0    1    0    0    0    0    1    0    0    0    2    0    1    1    1    1    1    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.16: Tek katmanlı 50 nörona sahip ağın Simülasyon çıktıları

```
>> [vcikti,vld_hedef] = siniragi2(n1,n2,loran, training_oran)

vcikti =

    1    1    0    0    0    1    0    0    1    0    0    0    1    1    0    1    0    1    1    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.17: Çift katmanlı 20-20 nörona sahip ağın Simülasyon çıktıları

```
>> [vcikti,vld_hedef] = siniragi2(n1,n2,loran, training_oran)

vcikti =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    2    1    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.18: Çift katmanlı 50-20 nörona sahip ağın Simülasyon çıktıları

Aynı verilerin BFGS quasi-Newton backpropagation öğrenme fonksiyonuyla eğitilmesi sonucu simüle edilen ağın validasyon çıktıları Şekil.4.1.3.6'da verilmiştir.

```
>> [vcikti,vld_hedef] = siniragi2(n1,n2,loran, training_oran)

vcikti =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    0    1    0    1    1    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.19: BFGS quasi-Newton backpropagation öğrenme fonksiyonuyla eğitilmesi sonucu simüle edilen ağın validasyon çıktıları

Aynı verilerin One step secant backpropagation fonksiyonuyla eğitilmesi sonucu simüle edilen ağın validasyon çıktıları Şekil.4.1.3.6’da verilmiştir.

```
>> [vcikti,vld_hedef] = siniragi2(n1,n2,loran, training_oran)

vcikti =

    0    1    0    1    1    1    0    0    1    0    0    0    1    1    0    1    1    1    0    0

vld_hedef =

    1    1    0    1    0    1    0    0    1    0    0    0    1    1    1    1    1    1    1    0
```

Şekil 3.20: One step secant backpropagation fonksiyonuyla eğitilmesi sonucu simüle edilen ağın validasyon çıktıları

Bölüm 4

SONUÇLAR ve TARTIŞMA

Bu çalışmada aç-ivme sensörü bir kaleme entegre edilerek kişilerin el titreme değerlerinin elde edilmesi ve bu değerlere bakılarak Parkinson hastalığına dair bir ön tanıda bulunmaya çalışılmıştır. Oluşturulan kalemle farklı şiddetle titreme değerleri içeren veri seti oluşturulmuştur. Tanı işlemi için makine öğrenme yöntemleri kullanılmıştır. Veri seti yapay sinir ağı eğitiminde kullanıldı. Eğitilen ağ, tanı yapılacak kişinin çizimi sırasında kaydedilen değerlerle simüle edilerek hastalık var veya yok olacak şekilde çıktı vermesi sağlanmıştır. Yapay sinir ağının eğitiminde kullanılacak olan, çizim esnasında sensörden kaydedilen ivme değeri sayısı ilk olarak 100 seçildi, her bir ölçümde 100 değer olan veri setiyle eğitilen yapay sinir ağının öğrenme oranının çok yüksek olduğu gözlemlendi. Değer sayısı 25'e indirildiğinde öğrenme oranı %80-90, 10'a indirildiğinde ise %70-79 oranında olduğu tespit edildi.

Yapay sinir ağının öğrenme-simüle başarısını etkileyen bir diğer faktör de ağ yapısında kullanılan nöron katman ve nöron sayısıdır. Yapay sinir ağını tek katmanlı 20 nöron ve 50 nöron kullanarak oluşturduğumuzda öğrenme oranını %90-99 olarak belirlerken, 10 nöron, 80 nöron ve 100 nöron kullanarak oluşturduğumuzda ise ortalama %70-80 oranına düştüğü gözlemlendi. Yapay sinir ağını 2 katmanlı 20-20 nöron, 30-10 nöron ve 50-20 nöron kullanarak oluşturduğumuzda ise öğrenme oranının %90-99 olduğu gözlemlendi.

Çalışma çizimlerdeki sapma miktarlarının tespiti, tanı konulacak

kişinin kalemi kavrama şiddeti ve bu şiddetin devamlılığı gibi faktörlerde eklenerek geliştirilebilir.

Ek A

MATLAB KODLARI

Tezinize MATLAB kodlarını eklemek için bu kısma ait koda bakınız.

```

1  %Arduino aci-ivme olcum
2  void setup() {
3  Wire.begin();
4  Serial.begin(38400);
5  Serial.println("I2C cihazlar baslatiliyor...");
6  accelgyro.initialize();
7  Serial.println("Test cihazı baglantıları...");
8  Serial.println(accelgyro.testConnection() ? "MPU6050 baglanti ...
    basarili" : "MPU6050 baglantisi basarisiz");
9  }
10 void loop() {
11 accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
12 Serial.print("a/g:\t");
13 Serial.print(ax); Serial.print("\t");
14 Serial.print(ay); Serial.print("\t");
15 Serial.print(az); Serial.print("\t");
16 Serial.print(gx); Serial.print("\t");
17 Serial.print(gy); Serial.print("\t");
18 Serial.println(gz)
19     }

1  %Arduino ana kodlar
2  #include "Wire.h"
3  #include "I2Cdev.h"
4  #include "MPU6050.h"
5  #include "SoftwareSerial.h"
6

```

```

7 MPU6050 mpu;
8
9 int16_t ax, ay, az;
10 int16_t gx, gy, gz;
11
12
13 long val, val2, val3;
14 long prevVal;
15 int buton=2;
16 int yesil=5;
17 int kirmizi=6;
18 int butondurum=0;
19 void setup()
20 {
21     Wire.begin();
22     Serial.begin(115200);
23     mpu.initialize();
24     pinMode(buton, INPUT_PULLUP);
25     pinMode(yesil, OUTPUT);
26     pinMode(kirmizi, OUTPUT);
27 }
28
29 void loop()
30 {
31     butondurum=digitalRead(buton);
32     if(butondurum==0){
33         digitalWrite(yesil, 0);
34         digitalWrite(kirmizi, 1);
35         mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
36         val = map(ax, -17000, 17000, -500, 500);
37         val2 = map(ay, -17000, 17000, -500, 500);
38         val3 = map(az, -17000, 17000, -500, 500);
39         Serial.println(val);
40         delay(50);
41         Serial.println(val2);
42         delay(50);
43         Serial.println(val3);
44         delay(50);
45     }
46     else{
47         digitalWrite(yesil, 1);
48         digitalWrite(kirmizi, 0);
49     }
50 }

1 clc;
2 clear all;

```

```

3
4  comport = serial('COM5','BaudRate',115200);
5  fopen(comport);
6  x=0;
7  while(x<100)
8  x=x+1;
9  y1(x)=fscanf(comport, '%d');
10 y2(x)=fscanf(comport, '%d');
11 y3(x)=fscanf(comport, '%d');
12 plot(y1, 'r', 'linewidth',1)
13 hold on;
14 plot(y2, 'b', 'linewidth',1)
15 plot(y3, 'g', 'linewidth',1)
16
17 hold off
18 drawnow
19 end
20 fclose(comport);
21 delete(comport);

1  %Sensor verilerinin kayıt edilmesi
2  comport = serial('COM5','BaudRate',115200);
3  fopen(comport);
4  x=0;
5  while(x<100)
6  x=x+1;comport = serial('COM5','BaudRate',115200);
7  fopen(comport);
8  x=0;
9  y1(x)=fscanf(comport, '%d');
10 y2(x)=fscanf(comport, '%d');
11 y3(x)=fscanf(comport, '%d');
12 end
13 f1=[y1(); y2();y3()];
14 a=f1.^2;
15 b=a(1,:)+a(2,:)+a(3,:);
16 c=sqrt(b);
17 ort2=c;
18 fclose(comport);
19 dosya=fopen('file1.txt','w');

1  %Sinir Agi olusturma ve egitilmesi
2  function[vcikti,vld_hedef] = siniragi2(n1,n2,loran, training_oran)
3  %input= sensor degerleri
4  %output=hastalik durumlari
5  %training_oran=Verilerin yuzde kacinin training icin ...
    kullanilacagini belirleme orani.

```

```

6 %Girdiyi ve ciktilarini training ve validation olarak ayir
7 %NN olusturup training verileriyle agi egitecegiz.
8 %Validation verilerini aga girdi olarak verecegiz. Egitilmis ...
    agdan cikan
9 %ciktilarla gercek ciktilari karsilastiracagiz
10 % Verilerin training ve validation ayrimi yapilir
11 verisayisi=size(giris,1);
12 trnsayi=round(verisayisi*training_oran);
13 trn_girdi=giris(1:trnsayi,:);
14 vld_girdi=giris(trnsayi+1:end,:);
15
16 trn_hedef=cikis(1:trnsayi,:);
17 vld_hedef=cikis(trnsayi+1:end);
18 %Verilerin sutunlarda olmasi icin transpozunu almaliyiz
19 trn_girdi=trn_girdi';
20 vld_girdi=vld_girdi';
21 trn_hedef=trn_hedef';
22 vld_hedef=vld_hedef';
23 %loran=learning rate yani ogrenme orani
24 net=newff(trn_girdi, trn_hedef, [n1,n2], {}, 'trainlm');
25 net.trainParam.lr=loran;
26 net.trainParam.epochs=1000;
27 net.trainParam.goal=1e-20;
28 net.trainParam.show=NaN; %Sonuclari train bitinceye kadar gosterme
29 %Agi egitme
30 net=train(net,trn_girdi,trn_hedef);
31 %Simule etme
32 vcikti=sim(net,vld_girdi);
33 vcikti=round(vcikti);
34 end

1 %tanida kullanılan sinir agi fonksiyonu
2 function[vcikti] = siniragi(giris,cikis,ort2,n1,n2,loran)
3 %input= sensor degerleri
4 %output=hastalik durumlari
5 %training_oran=Verilerin yuzde kacinin training icin ...
    kullanicagini belirleyen oran.
6 %Girdiyi ve ciktilari training ve validation olarak ayir
7 %NN olusturup training verileriyle agi egitecegiz.
8 %Validation verilerini aga girdi olarak verecegiz. Egitilmis ...
    agdan cikan
9 %ciktilarla gercek ciktilari karsilastiracagiz
10 % Verilerin training ve validation ayrimi yapilir
11 trn_girdi=giris();
12 vld_girdi=ort2;
13 trn_hedef=cikis();
14

```

```

15 %Verilerin sutunlarda olmasi icin transpozunu almaliyiz
16 trn_girdi=trn_girdi';
17 vld_girdi=vld_girdi';
18 trn_hedef=trn_hedef';
19
20 %loran=learning rate
21 net=newff(trn_girdi, trn_hedef, [n1,n2], {}, 'trainlm');
22 net.trainParam.lr=loran;
23 net.trainParam.epochs=1000;
24 net.trainParam.goal=1e-25;
25 net.trainParam.show=NaN; %Sonuclari train bitinceye kadar gosterme
26 %Agi egitme
27 net=train(net,trn_girdi,trn_hedef);
28 %Simule etme
29 vcikti=sim(net,vld_girdi);
30 if vcikti<0.5
31     vcikti=0;
32 else
33     vcikti=1;
34 end
35 end

```

```

1 %On tanida kullanılan ana algoritma
2 clc;
3 clear all;
4 comport = serial('COM5','BaudRate',115200);
5 fopen(comport);
6 x=0;
7 while(x<25)
8     x=x+1;
9     y1(x)=fscanf(comport, '%d');
10    y2(x)=fscanf(comport, '%d');
11    y3(x)=fscanf(comport, '%d');
12 end
13 f1=[y1(); y2();y3()];
14 a=f1.^2;
15 b=a(1,:)+a(2,:)+a(3,:);
16 ort2=sqrt(b);
17 fclose(comport);
18 n1=30;
19 n2=20;
20 loran=0.8;
21 [vcikti] = siniragi(giris,cikis,ort2,n1,n2,loran)

```

KAYNAKÇA

- [1] "Türk Nöroloji Derneği, “ <http://www.noroloji.org.tr/parkinson%20hastaligi.pdf>,” , 2014.
- [2] Arduino Playground, “ <https://playground.arduino.cc/Main/MPU-6050>,” , 2020.
- [3] İbrahim Çayroğlu, “<http://www.ibrahimcayiroglu.com/IleriAlgoritmaAnalizi-YapaySinirAglari.pdf>,” , 2020.