



**T.C.**

**YALOVA ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YAPAY SİNİR AĞLARI DERSİ PROJE ÖDEVİ**

**YAPAY SİNİR AĞI MODELİ İLE BEYZBOL MAAŞ TAHMİNİ**

**HAZIRLAYAN**

**200101061 - DENİZ YAĞMUR SOYLU**

**DANIŞMAN: GÜNEŞ HARMAN**

**YALOVA, 2024**

## İÇİNDEKİLER

1) YAPAY SİNİR AĞLARI NEDİR? .....	1
1.1) Yapay Sinir Ağlarının Avantajları .....	1
1.2) Yapay Sinir Ağlarının Kullanım Alanları .....	2
1.3) Yapay Sinir Ağlarının Çalışma Prensibi .....	2
1.4) Bir Yapay Sinir Hücresinin Bölümleri.....	3
2) YAPAY SİNİR AĞLARI İLE BEYZBOL MAAŞ TAHMİNİ PROJESİ .....	4
2.1) Veri Seti Hikayesi .....	4
2.2) Eksik Verilerin Doldurulması.....	5
2.3) Kategorik Verilerin Dönüştürülmesi.....	6
2.4) Eğitim ve Test Verilerinin Ayrılması .....	6
2.5) Verilerin Standartlaştırılması.....	6
2.6) Yapay Sinir Ağı Modelinin Kurulması .....	6
2.7) Ağırlık ve Biasların Başlangıç Değerleri .....	7
2.8) Modelin Derlenmesi ve Eğitilmesi .....	7
2.9) Modelin Değerlendirilmesi.....	7
2.10) Sonuçların Görselleştirilmesi .....	8
3) PROJE ÇIKTILARI VE YORUMLANMASI .....	9
3.1) ReLU Aktivasyon Fonksiyonu .....	9
3.1.1) ReLU Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu .....	11
3.2) Sigmoid Aktivasyon Fonksiyonu .....	11
3.2.1) Sigmoid Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu .....	13
3.3) Tanh (Hiperbolik Tanjant) Aktivasyon Fonksiyonu.....	13
3.3.1) Tanh Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu .....	15
Genel Değerlendirme.....	15
KAYNAKÇA.....	16

## KISALTMA LİSTESİ

LOSS	Hata Ortalaması
MAE	Ortalama Mutlak Hata
MLP	Çok Katmanlı Perceptron
MSE	Küçük Hata Kareler Ortalaması
$R^2$	R-Karesi
ReLU	Rectified Linear Unit
RNN	Tekrarlayan Sinir Ağları
TANH	Hiperbolik Tanjant
YSA	Yapay Sinir Ağları

## ÖNSÖZ

Bu proje, yapay sinir ağıları (YSA) kullanarak beyzbol oyuncularının maaşlarını tahmin etmeyi amaçlayan bir çalışmadır. Yapay sinir ağıları, günümüzün en güçlü makine öğrenimi tekniklerinden biri olarak, karmaşık veri setlerini analiz etme ve doğru tahminlerde bulunma konusunda büyük bir potansiyele sahiptir. Bu projede, yapay sinir ağlarının temel prensiplerini öğrenmek, model kurma sürecini anlamak ve farklı aktivasyon fonksiyonlarının etkilerini incelemek amacıyla, gerçek bir veri seti üzerinde çalışılmıştır.

Yapay sinir ağlarının, insan beyninin çalışma prensiplerinden esinlenerek oluşturulan bir yapay zeka tekniği olduğunu ve bu sistemlerin verilerdeki gizli ilişkileri öğrenebilme yeteneğine sahip olduğunu bu projede derinlemesine keşfetme fırsatı bulduk. Ayrıca, farklı aktivasyon fonksiyonlarının kullanıldığı modelleri test ederek, her bir fonksiyonun maaş tahmini üzerindeki etkilerini inceledik ve bu aktivasyon değerlerinden çıkan sonuçları karşılaştırdık.

Proje ödevimizle ilgili rehberliği ve destekleri için Dr. Öğr. Gör. Güneş Harman hocamıza içtenlikle teşekkür ederiz. Onun katkıları ve rehberliği sayesinde, yapay sinir ağları hakkında derinlemesine bilgi edinmek ve bu bilgileri uygulamaya koymak mümkün olmuştur. Bu çalışma, yapay sinir ağlarının işleyişini anlamaya ve gerçek dünyadaki uygulamalarını görmeye yönelik bir adım atmak isteyen herkes için değerli bir kaynak oluşturmayı amaçlamaktadır.

## ŞEKİL LİSTESİ

Şekil 1.1 Bir Nöronun Yapay Sinir Ağı ile İlişkisi

Şekil 1.2 Yapay Sinir Ağlarının Çalışma Prensibi

Şekil 1.3 Yapay Sinir Ağları Katmanları

Şekil 2.1 Beyzbol Maaş Tahmin Projesi

Şekil 2.2 Yapay Sinir Ağı Modeli

Şekil 2.3 Modele Ağırlık ve Bias Değerlerinin Atanması

Şekil 3.1 ReLU Aktivasyon Fonksiyonu Grafiği

Şekil 3.2 Sigmoid Aktivasyon Fonksiyonu Grafiği

Şekil 3.3 Tanh Aktivasyon Fonksiyonu Grafiği

## **TABLO LİSTESİ**

Tablo 3.1 ReLU Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

Tablo 3.2 Sigmoid Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

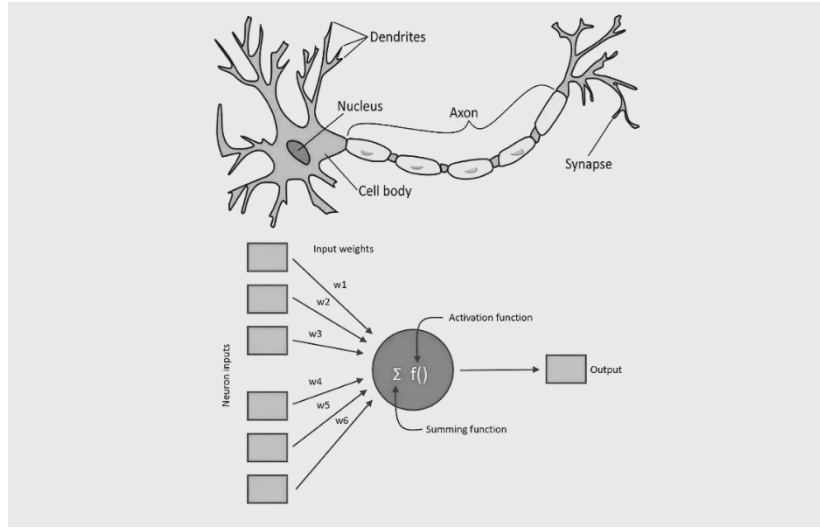
Tablo 3.3 Tanh Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

## ÇİZELGE LİSTESİ

Çizelge 2.1 Hitters Veri Setindeki Değişkenler ve Açıklamaları

## 1. YAPAY SİNİR AĞLARI NEDİR?

Yapay sinir ağı (YSA), insan beyninin öğrenme, yeni bilgiler üretme ve keşfetme gibi özelliklerini otomatik bir şekilde taklit etmeyi amaçlayan bilgisayar sistemleridir. Bu sistemler, insan beyninin işleyişini model alarak, matematiksel bir öğrenme süreciyle geliştirilmiştir. Biyolojik sinir ağlarının yapısını, öğrenme, hatırlama ve genelleme yeteneklerini örnek alır. Öğrenme süreci sırasında, yapay sinir ağına örnekler verilerek giriş ve çıkışlar arasında ilişkiler kurulması sağlanır. Biyolojik nöronlarda dendritler girişleri alırken, yapay nöronlarda giriş verileri ve ağırlıklarla hesaplama yapılır. Biyolojik nöronun çıktısı sinapslarla diğer hücrelere bağlanırken, yapay nöronun çıktısı bir sonraki katmana aktarılır (Şekil 1.1).



Şekil 1.1 Bir Nöronun Yapay Sinir Ağı ile İlişkisi

### 1.1 Yapay Sinir Ağlarının Avantajları

- Yapay sinir ağı, birçok hücreden oluşur ve bu hücreler eş zamanlı çalışarak karmaşık problemleri çözebilir.
- Farklı öğrenme algoritmaları sayesinde öğrenme kabiliyetine sahiptir.
- Daha önce karşılaşılmamış durumlar için sonuç üretebilir (gözetimsiz öğrenme).
- Örüntü tanıma, sınıflandırma ve eksik örüntüleri tamamlama gibi işlemleri gerçekleştirebilir.
- Eksik ya da belirsiz verilerle çalışabilir ve hata durumlarında dereceli bozulma (graceful degradation) gösterir.
- Paralel çalışabilme özelliğiyle gerçek zamanlı veri işleyebilir.



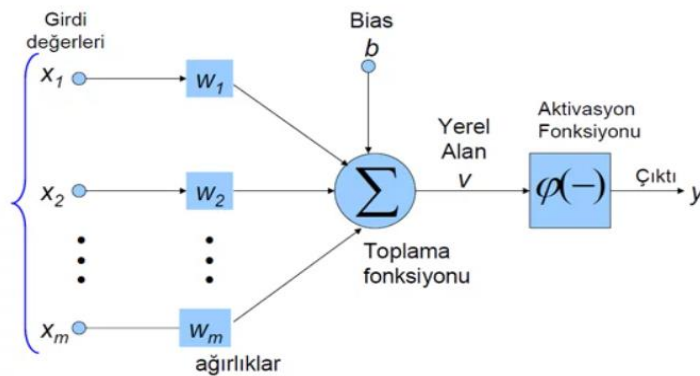
Yapay sinir ağıları; teşhis, sınıflandırma, tahmin, kontrol, veri ilişkilendirme, filtreleme ve yorumlama gibi alanlarda yaygın olarak kullanılmaktadır. Hangi problem için hangi ağ yapısının uygun olduğunu belirlemek, problemin özelliklerini ve ağın yeteneklerini karşılaştırmayı gerektirir.

## 1.2 Yapay Sinir Ağlarının Kullanım Alanları

- Hesaplamalı Finans: Kredi skorlaması, algoritmik ticaret.
- Görüntü İşleme ve Bilgisayarla Görü: Yüz tanıma, hareket algılama, nesne tespiti.
- Hesaplamalı Biyoloji: Tümör tespiti, ilaç keşfi, DNA dizilimi.
- Enerji Üretimi: Fiyat ve yük tahmini.
- Otomotiv, Havacılık ve Üretim: Öngörücü bakım.
- Doğal Dil İşleme: Sesli asistanlar, duygu analizi.

## 1.3 Yapay Sinir Ağlarının Çalışma Prensibi

Dendrit adı verilen yollar boyunca belirli ağırlıklar bulunmaktadır ve bu dendritlere bir başka nörondan gelen bir giriş değeri ulaşabilir. Giriş değeri ile dendritteki ağırlık çarpıldıktan sonra sinir hücresine iletilir. Sinir hücresinde bu çarpma işlemi tüm dendritlerden gelen girişlerle tekrarlanır ve ağırlıklı toplam elde edilir. Bu toplam, bir bias değeri ile birleştirildikten sonra aktivasyon fonksiyonundan geçirilir ve bir çıkış üretilir. Bu çıkış, nihai sonuç olabileceği gibi başka bir hücrenin girdisi olarak da kullanılabilir. Matematiksel olarak bu işlem, ağırlıkların girişlerle çarpılması ve üzerine bir bias eklenmesiyle ifade edilir. Böylece basit bir matematiksel model oluşturulur (Şekil 1.2).



Şekil 1.2 Yapay Sinir Ağlarının Çalışma Prensibi

Yapay Sinir Ağlarında temel amaç, modelin en iyi performansı vermesini sağlayacak şekilde ağırlık ve bias parametrelerini optimize etmektir.

#### 1.4 Bir Yapay Sinir Hücresinin Bölümleri

Her sinir hücresi aynı işlem prensibiyle çalışır ve hücreler birbirine seri veya paralel şekilde bağlanarak yapıyı oluşturur (Şekil 1.3). Bir yapay sinir hücresi temel olarak beş bileşenden oluşur:

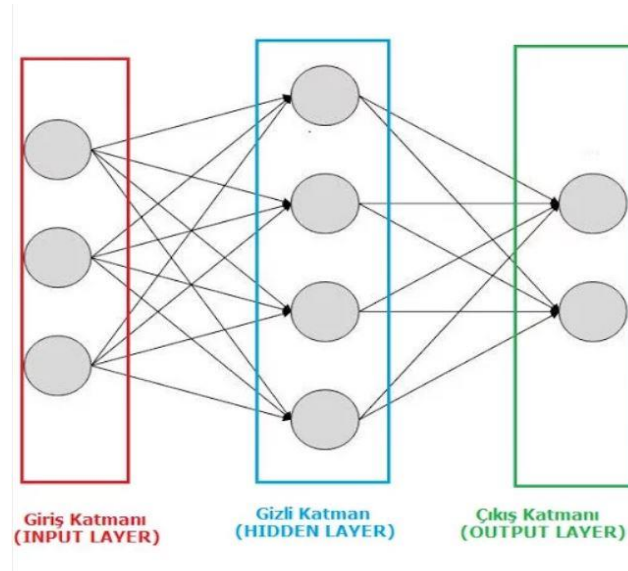
**Girdiler:** Nörona gelen veriler, biyolojik sinir hücrelerindeki gibi çekirdeğe iletmek üzere toplanır.

**Ağırlıklar:** Hücreye gelen bilgiler, çekirdeğe ulaşmadan önce bağlı oldukları bağlantıların ağırlıklarıyla çarpılır. Böylece her girdinin üretilen çıktı üzerindeki etkisi ayarlanabilir.

**Toplama Fonksiyonu:** Bu fonksiyon, ağırlıklarla çarpılarak gelen girdileri toplar ve hücrenin net girişini hesaplar.

**Aktivasyon Fonksiyonu:** Ağırlıklı toplam değerini doğrusal olmayan bir formda dönüştürür ve bu sonucu bir sonraki katmana iletir (örneğin, ReLU veya sigmoid).

**Çıktılar:** Aktivasyon fonksiyonundan elde edilen sonuç hücrenin çıktısıdır. Her hücre birden fazla giriş alabilir, ancak yalnızca bir çıkış üretir. Bu çıktı, istenilen sayıda başka hücreye bağlanabilir.



Şekil 1.3 Yapay Sinir Ağları Katmanları

## 2. YAPAY SİNİR AĞLARI İLE BEYZBOL MAAŞ TAHMİNİ PROJESİ

Bu projede, Hitters veri seti kullanılarak beyzbol oyuncularının maaşlarını tahmin etmek hedeflenmektedir (Şekil 2.1). Proje, oyuncuların istatistiksel performansı ve diğer özelliklerini girdi olarak kullanıp, bir yapay sinir ağı (YSA) modeli ile maaş tahmini yapmayı hedeflemektedir. Aşağıda, proje adımları detaylandırılmıştır.



Şekil 2.1 Beyzbol Maaş Tahmin Projesi

### 2.1 Veri Seti Hikayesi

Bu projede kullanılacak olan Hitters adlı veri seti Kaggle'dan çekilmektedir ve pandas kütüphanesi kullanılarak işlenmektedir. Veri seti, beyzbol oyuncularının performansları ve maaşlarına ilişkin çeşitli bilgileri içermektedir. Bu veri setinde toplam 20 değişken ve 322 adet gözlem bulunmaktadır (Çizelge 2.1).

DEĞİŞKEN	AÇIKLAMASI
AtBat	1986-1987 sezonunda bir beyzbol sopası ile topa yapılan vuruş sayısı
Hits	1986-1987 sezonundaki isabet sayısı
HmRun	1986-1987 sezonundaki en değerli vuruş sayısı
Runs	1986-1987 sezonunda takımına kazandırdığı sayı
RBI	Bir vurucunun vuruş yaptığında koşu yaptırdığı oyuncu sayısı

<b>Walks</b>	Karşı oyuncuya yaptırılan hata sayısı
<b>Years</b>	Oyuncunun majör liginde oynama süresi (sene)
<b>CAtBat</b>	Oyuncunun kariyeri boyunca topa vurma sayısı
<b>CHits</b>	Oyuncunun kariyeri boyunca yaptığı isabetli vuruş sayısı
<b>CHmRun</b>	Oyuncunun kariyeri boyunca yaptığı en değerli sayısı
<b>CRuns</b>	Oyuncunun kariyeri boyunca takımına kazandırdığı sayı
<b>CRBI</b>	Oyuncunun kariyeri boyunca koşu yaptırtıldığı oyuncu sayısı
<b>CWalks</b>	Oyuncun kariyeri boyunca karşı oyuncuya yaptırdığı hata sayısı
<b>League</b>	Oyuncunun sezon sonuna kadar oynadığı ligi gösteren A ve N seviyelerine sahip bir faktör
<b>Division</b>	1986 sonunda oyuncunun oynadığı pozisyonu gösteren E ve W seviyelerine sahip bir faktör
<b>PutOuts</b>	Oyun içinde takım arkadaşınla yardımlaşma
<b>Assits</b>	1986-1987 sezonunda oyuncunun yaptığı asist sayısı
<b>Errors</b>	1986-1987 sezonundaki oyuncunun hata sayısı
<b>Salary</b>	Oyuncunun 1986-1987 sezonunda aldığı maaş(bin üzerinden)
<b>NewLeague</b>	1987 sezonunun başında oyuncunun ligini gösteren A ve N seviyelerine sahip bir faktör

Çizelge 2.1 Hitters Veri Setindeki Değişkenler ve Açıklamaları

## 2.2 Eksik Verilerin Doldurulması

Veri setinde yalnızca bağımlı değişken olan "Salary" sütununda eksik veriler tespit edilmiş olup bu eksik değerler, sütunun medyan değeriyle doldurulmuştur. Bu sayede, eksik verilerin modeli olumsuz etkilemesinin önüne geçilmiştir.

### 2.3 Kategorik Verilerin Dönüştürülmesi

Veri setinde bulunan "League" (Oyuncunun sezon sonuna kadar oynadığı ligi gösteren A ve N seviyelerine sahip bir faktör), "Division" (1986 sonunda oyuncunun oynadığı pozisyonu gösteren E ve W seviyelerine sahip bir faktör) ve "NewLeague" (1987 sezonunun başında oyuncunun ligini gösteren A ve N seviyelerine sahip bir faktör ) gibi kategorik sütunlar, one-hot encoding yöntemi ile sayısal formatlara dönüştürülmüştür. Bu adım, oluşturulan modelin kategorik verilerle daha etkin çalışmasını sağlamak için uygulanmıştır.

### 2.4 Eğitim ve Test Verilerinin Ayrılması

Veriler, %80 eğitim ve %20 test oranıyla ikiye ayrılmıştır. Bu işlem için `train_test_split()` fonksiyonu kullanılmıştır. Ayrım, modelin performansını bağımsız bir test seti üzerinde değerlendirebilmek için gereklidir.

### 2.5 Verilerin Standartlaştırılması

Veriler, `StandardScaler` kullanılarak normalize edilmiştir. Bu işlem, verilerin ortalamasını 0, standart sapmasını ise 1 olacak şekilde dönüştürmeyi sağlar. Normalize edilmiş veriler, yapay sinir ağının eğitim sürecinde daha iyi performans göstermesine yardımcı olur.

### 2.6 Yapay Sinir Ağı Modelinin Kurulması

Model, TensorFlow ve Keras kütüphaneleri kullanılarak oluşturulmuştur. Modelin yapısı şu şekildedir (Şekil 2.2):

- **Giriş Katmanı:** 32 nöron, ReLU aktivasyon fonksiyonu.
- **Gizli Katman 1:** 16 nöron, ReLU aktivasyon fonksiyonu.
- **Çıkış Katmanı:** 1 nöron (doğrusal).

```
# Yapay sinir ağı modelini oluşturma
model = Sequential([
    Dense(32, input_dim=X_train_scaled.shape[1], activation='relu'),
    Dense(16, activation='relu'),
    Dense(1)
])
```

Şekil 2.2 Yapay Sinir Ağı Modeli

## 2.7 Ağırlık ve Biasların Başlangıç Değerleri

Modelin ilk katmanında bulunan ağırlık ve bias değerleri, rastgele bir aralık (0.0001 ile 0.08 arası) içinde atanarak başlatılmıştır (Şekil 2.3). Bu adım, modelin eğitime daha iyi başlamasını sağlamaktadır.

```
custom_weights = np.random.uniform(0.0001, 0.08, size=(X_train_scaled.shape[1], 32)) # 16 units
custom_biases = np.random.uniform(0.01, 0.05, size=(32,)) # 16 biases for 32 units

# Ağırlıkları ve biasları ilk katmana ayarlama
model.layers[0].set_weights([custom_weights, custom_biases])
```

Şekil 2.3 Modele Ağırlık ve Bias Değerlerinin Atanması

Modelin parametreleri (ağırlık ve bias değerleri), rastgele değerlerle başlatılmış ve ilk katman için özelleştirilmiştir.

## 2.8 Modelin Derlenmesi ve Eğitilmesi

Model, aşağıdaki parametreler ile derlenmiştir:

- **Optimizasyon Algoritması:** Adam algoritması, modelin ağırlıklarını optimize etmek için kullanılmıştır.
- **Kayıp Fonksiyonu:** Küçük hata kareler ortalaması (MSE), modelin tahmin performansını ölçmek için tercih edilmiştir.
- **Erken Durdurma Mekanizması:** Modelin doğrulama setindeki MAE (Mean Absolute Error) değerinin 200 epoch boyunca iyileşmemesi durumunda eğitim süreci sonlandırılmış ve en iyi ağırlıklar geri yüklenmiştir.

Eğitim süreci, 32 batch boyutuyla toplam 300 epoch boyunca gerçekleştirilmiştir. Bu süreç boyunca modelin hem eğitim hem de doğrulama setleri üzerindeki performansı düzenli olarak izlenmiştir.

## 2.9 Modelin Değerlendirilmesi

Test seti kullanılarak modelin performansı değerlendirilmiştir. Performans sonucunda hesaplanan metrikler şu şekildedir:

- **Hata Ortalaması (Loss):** Modelin genel hata seviyesini gösterir.

- **Ortalama Mutlak Hata (MAE):** Model tahminlerinin yanlışlık seviyesini belirtir.
- **R-Karesi ( $R^2$ ):** Modelin açıklama gücünü ifade eder.

Test setinde hesaplanan  $R^2$  skoru, modelin tahmin kabiliyetini ölçmüştür.

## 2.10 Sonuçların Görselleştirilmesi

Sonuçların daha anlaşılır hale gelmesi için bazı grafikler oluşturulmuştur:

- **Eğitim ve Doğrulama Kayıpları:** Epoch bazında eğitim ve doğrulama kayıplarını karşılaştıran grafiklerdir.
- **Gerçek ve Tahmin Değerleri:** Test setindeki gerçek maaşlarla tahmin edilen maaşları karşılaştıran bir dağılım grafiğidir.
- **Hata Histogramı:** Tahmin hatalarını (Gerçek - Tahmin) oranı ile gösteren bir histogramdır.
- **Kümülatif Dağılım Grafiği:** Tahmin hatalarının kümülatif dağılımını gösteren grafiklerdir.

Bu adımlar sonucunda geliştirilen yapay sinir ağı modeli, beyzbol oyuncularının maaşlarını tahmin etmek için başarılı bir performans sergilemiştir. Model, veri setindeki karmaşık ilişkileri anlamış ve doğruluk seviyesini önemli ölçüde artırmıştır.

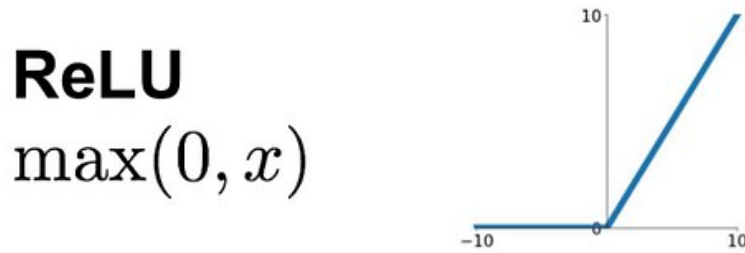
### 3. PROJE ÇIKTILARI VE YORUMLANMASI

#### 3.1 ReLU Aktivasyon Fonksiyonu

ReLU (Rectified Linear Unit), derin öğrenme ve yapay sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. ReLU, giriş değerini doğrudan geçiren ve negatif değerleri sıfıra eşitleyen basit bir fonksiyondur. Matematiksel olarak, ReLU fonksiyonu şu şekilde tanımlanır:

$$f(x) = \max\{0, x\}$$

Bu ifade, x pozitif olduğunda f(x) değerinin x olduğunu, negatif olduğunda ise f(x) değerinin 0 olduğunu gösterir (Şekil 3.1).



Şekil 3.1 ReLU Aktivasyon Fonksiyonu Grafiği

#### Özellikler

- 1. Basitlik:** ReLU fonksiyonu, hesaplama açısından oldukça basittir. Bu, modelin daha hızlı eğitilmesine olanak tanır.
- 2. Doğrusal Olmayanlık:** Doğrusal bir aktivasyon fonksiyonu kullanmak, modelin yalnızca doğrusal ilişkileri öğrenmesine neden olur. ReLU, doğrusal olmayan bir yapı sağlayarak modelin karmaşık ilişkileri öğrenmesine yardımcı olur.
- 3. Sparsity (Seyreklik):** ReLU'nun negatif girişleri sıfıra eşitlemesi, bazı nöronların devre dışı kalmasına neden olur. Bu durum, modelin daha seyrek (sparse) hale gelmesini sağlar ve bu da genellikle daha iyi genelleme sağlar.
- 4. Hızlı Konverjans:** ReLU, diğer aktivasyon fonksiyonlarına göre daha hızlı bir şekilde konverjans sağlar. Bu, özellikle derin sinir ağlarında eğitim süresini önemli ölçüde kısaltabilir.



## Avantajları

- **Hızlı Hesaplama:** ReLU'nun basit matematiksel formülü sayesinde hesaplamalar hızlıdır.
- **Aşırı Öğrenmeye Karşı Dayanıklılık:** ReLU, aşırı öğrenmeyi azaltmaya yardımcı olabilir çünkü bazı nöronlar sıfıra eşitlenerek devre dışı kalır.
- **Derin Ağlar için Uygunluk:** Derin sinir ağlarında kullanılmak üzere tasarlanmıştır ve genellikle iyi sonuçlar verir.

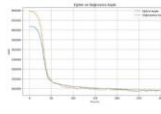
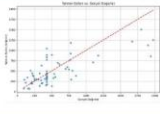
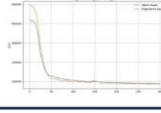
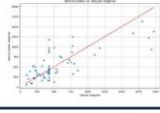
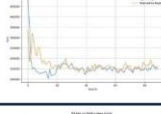
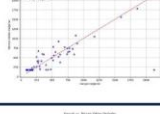
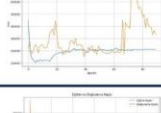
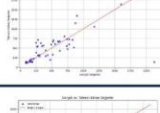
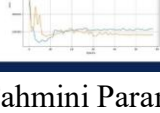

## Dezavantajları

- **Ölü Nöron Problemi:** Eğer bir nöron sürekli olarak negatif giriş alıyorsa, bu nöron "ölü" hale gelir ve eğitim sırasında güncellenmez. Bu durum, modelin kapasitesini azaltabilir.
- **Düşük Geri Yayılım:** Negatif girişlerde geri yayılım sırasında gradyanlar sıfır olduğundan, bu durum bazı nöronların öğrenmesini engelleyebilir.

## Kullanım Alanları

ReLU, çoğu derin öğrenme uygulamasında yaygın olarak kullanılır. Özellikle görüntü işleme (CNN'ler), doğal dil işleme (RNN'ler) ve genel regresyon problemleri gibi birçok alanda tercih edilir.

### 3.1.1 ReLU Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

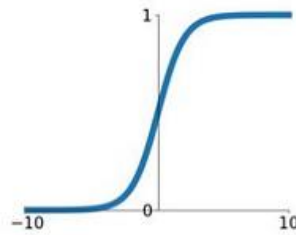
RELU AKTİVASYON FONKSİYONU											
BIAS BAŞLANGIÇ	BIAS BİTİŞ	AĞIRLIK BAŞLANGIÇ	AĞIRLIK BİTİŞ	GİRİŞ KATMANI	1.GİZLİ KATMAN	2.GİZLİ KATMAN	3.GİZLİ KATMAN	R-squared	Learning Rate	Eğitim ve Doğrulama Kaybı	Gerçek vs. Tahmin Edilen Değerler
0.001	0.1	0.001	0.1	16	16	4	2	0.5441	0.001		
0.0001	0.08	0.01	0.1	32	16	4	2	0.5645	0.1		
0.01	0.02	0.01	0.02	64	64	32	16	0.62	0.15		
0.01	0.02	0.01	0.02	512	512	256	128	0.66	0.1		
0.01	0.02	0.01	0.02	2048	2048	1024	512	0.78	0.4		

Tablo 3.1 ReLU Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

### 3.2) Sigmoid Aktivasyon Fonksiyonu

Sigmoid aktivasyon fonksiyonu, yapay sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Giriş değerlerini 0 ile 1 arasında bir çıktıya dönüştürür. Matematiksel olarak sigmoid fonksiyonu şu şekilde tanımlanır (Şekil 3.2):

$$\text{Sigmoid}$$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$


Şekil 3.2 Sigmoid Aktivasyon Fonksiyonu Grafiği

Burada e doğal logaritmanın tabanıdır ve x giriş değeridir. Sigmoid fonksiyonu, S şeklinde bir eğri oluşturur ve bu özellik, onu ikili sınıflandırma problemlerinde oldukça kullanışlı hale getirir.

## Özellikler

- **Çıktı Aralığı:** Sigmoid fonksiyonu, çıktısını her zaman 0 ile 1 arasında sınırlar. Bu, özellikle olasılık tahminleri için idealdir.
- **Türevlenebilirlik:** Sigmoid fonksiyonu, sürekli ve türevlenebilir bir fonksiyon olduğundan, geri yayılım algoritması ile öğrenme sürecinde kullanılabilir.
- **Monotonluk:** Fonksiyon monoton bir şekilde artar; yani, girdi değeri arttıkça çıktı da artar.

## Avantajları

- **Olasılık Çıktısı:** İkili sınıflandırma problemlerinde çıktıların olasılık olarak yorumlanabilmesi büyük bir avantajdır.
- **Sürekli ve Türevlenebilir:** Öğrenme süreçlerinde gradyan tabanlı yöntemler için uygundur.

## Dezavantajları

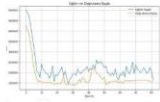
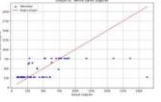
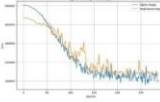
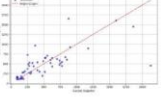
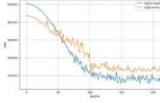
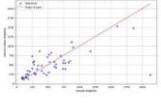
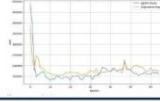
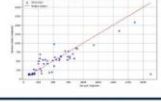
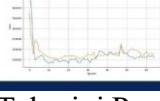
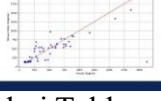
- **Kaybolan Gradyan Problemi:** Girdi değerleri çok büyük veya çok küçük olduğunda, sigmoid fonksiyonunun gradyanı sıfıra yaklaşır. Bu durum, ağırlık güncellemelerinin etkisiz hale gelmesine neden olur ve modelin öğrenme sürecini yavaşlatır.
- **Çıkışların Sınırlı Olması:** Çıktıların sadece 0 ile 1 arasında olması, bazı durumlarda modelin esnekliğini kısıtlayabilir.

## Kullanım Alanları

Sigmoid aktivasyon fonksiyonu, özellikle aşağıdaki alanlarda yaygın olarak kullanılır:

- **İkili Sınıflandırma:** İki sınıf arasında karar vermek için uygun bir aktivasyon fonksiyonu olarak kullanılır.
- **Çok Katmanlı Perceptron (MLP):** Sinir ağlarının gizli katmanlarında kullanılabilir.
- **Lojistik Regresyon:** İkili sonuçlar elde etmek için lojistik regresyon modellerinde sıkça tercih edilir.

### 3.2.2 Sigmoid Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

SİGMOİD AKTİVASYON FONKSİYONU											
BIAS BAŞLANGIÇ	BIAS BİTİŞ	AĞIRLIK BAŞLANGIÇ	AĞIRLIK BİTİŞ	giriş KATMANI	1.GİZLİ KATMAN	2.GİZLİ KATMAN	3.GİZLİ KATMAN	R-squared	Learning Rate	Eğitim ve Doğrulama Kaybı	Gerçek vs Tahmin Edilen Değerler
0.01	0.02	0.01	0.02	16	16	4	2	0.33	0.1		
0.001	0.005	0.001	0.005	32	32	16	4	0.46	0.01		
0.001	0.005	0.001	0.005	64	64	32	16	0.48	0.01		
0.01	0.02	0.0001	0.0005	128	128	64	32	0.47	0.01		
0.001	0.002	0.001	0.005	256	128	64	32	0.55	0.01		

Tablo 3.2 Sigmoid Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

### 3.3 Tanh (Hiperbolik Tanjant) Aktivasyon Fonksiyonu

Tanh (hiperbolik tanjant) aktivasyon fonksiyonu, yapay sinir ağlarında yaygın olarak kullanılan bir aktivasyon fonksiyonudur. Giriş değerlerini -1 ile 1 arasında bir aralığa dönüştürür. Matematiksel olarak tanh fonksiyonu şu şekilde tanımlanır (Şekil 3.3):



Şekil 3.3 Tanh Aktivasyon Fonksiyonu Grafiği

Bu formül, herhangi bir gerçek sayıyı -1 ile 1 arasındaki bir değere eşler.

#### Özellikler

- **Çıktı Aralığı:** Tanh fonksiyonu, çıktısını -1 ile 1 arasında sınırlar. Bu, negatif girişlerin güçlü bir şekilde negatif olarak eşleştirilmesine olanak tanır.

- **Sıfır Merkezli Çıkış:** Tanh, sıfır merkezli bir aktivasyon fonksiyonudur; bu, çıktıların ortalama değerinin sıfır olduğu anlamına gelir. Bu özellik, optimizasyon sürecini daha kolay hale getirebilir.
- **Türevlenebilirlik:** Tanh fonksiyonu sürekli ve türevlenebilir olduğundan, geri yayılım algoritması ile öğrenme sürecinde kullanılabilir.

### Avantajları

1. **Sıfır Merkezli Çıkış:** Sıfır merkezli olması, gradyanların sıfır etrafında daha simetrik dağılmasına yardımcı olur. Bu durum, öğrenme sürecini hızlandırabilir.
2. **Daha Güçlü Gradyanlar:** Tanh fonksiyonunun gradyanları, sigmoid fonksiyonuna göre daha diktir. Bu da kaybolan gradyan probleminin azaltılmasına yardımcı olur.

### Dezavantajları

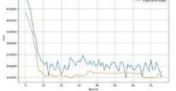
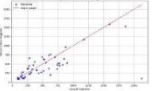
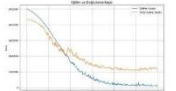
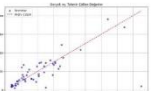
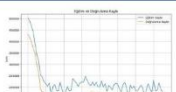



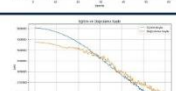

1. **Kaybolan Gradyan Problemi:** Tıpkı sigmoid fonksiyonunda olduğu gibi, çok büyük veya çok küçük giriş değerlerinde tanh fonksiyonunun gradyanı sıfıra yaklaşabilir. Bu durum, bazı nöronların öğrenmesini engelleyebilir.
2. **Hesaplama Maliyeti:** Sigmoid fonksiyonuna göre daha karmaşık bir hesaplama gerektirir; bu nedenle bazı uygulamalarda hesaplama verimliliği açısından dezavantajlı olabilir.

### Kullanım Alanları

Tanh aktivasyon fonksiyonu genellikle aşağıdaki alanlarda kullanılır:

- **Gizli Katmanlar:** Derin sinir ağlarının gizli katmanlarında yaygın olarak tercih edilir.
- **İkili Sınıflandırma:** İkili sınıflandırma problemlerinde kullanılabilir.
- **Dizisel Veriler:** RNN (Tekrarlayan Sinir Ağları) gibi dizisel verilerle çalışan modellerde sıkça kullanılır.

### 3.3.1 Tanh Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

TANH AKTİVASYON FONKSİYONU											
BIAS BAŞLANGIÇ	BIAS BİTİŞ	AĞIRLIK BAŞLANGIÇ	AĞIRLIK BİTİŞ	GİRİŞ KATMANI	1.GİZLİ KATMAN	2.GİZLİ KATMAN	3.GİZLİ KATMAN	R-squared	Learning Rate	Eğitim ve Doğrulama Kaybı	Gerçek vs Tahmin Edilen Değerler
0.001	0.002	0.001	0.005	16	16	4	2	0.41	0.1		
0.01	0.02	0.015	0.018	1024	1024	512	256	0.42	0.001		
0.001	0.002	0.001	0.005	32	32	16	4	0.45	0.01		
0.001	0.002	0.01	0.02	128	64	32	16	0.50	0.01		
0.01	0.02	0.01	0.02	64	64	32	16	0.62	0.01		

Tablo 3.3 Tanh Aktivasyon Fonksiyonu Kullanarak Maaş Tahmini Parametreleri Tablosu

#### Genel Değerlendirme

Gerçekleştirilen çalışmada, farklı aktivasyon fonksiyonları (ReLU, Tanh ve Sigmoid) kullanılarak maaş tahmini modeli geliştirilmiş ve performansları karşılaştırılmıştır. Model, giriş ve gizli katmanlar için çeşitli parametrelerle eğitilerek, R-squared ( $R^2$ ) skorları ve tahmin doğruluğu değerlendirilmiştir. Yapılan değerlendirmeler sonucunda en iyi  $R^2$  skoru **0.78** ile Relu aktivasyon fonksiyonu olarak bulunmuştur. Model, özellikle daha büyük ağ yapılarında (örneğin, daha fazla nöron ve gizli katman) daha başarılı sonuçlar vermiştir. Eğitim ve doğrulama kayıpları stabil şekilde azalarak, tahmin doğruluğunun yüksek olduğunu göstermiştir.

ReLU aktivasyon fonksiyonu hem öğrenme sürecindeki hızlı yakınsama hem de daha yüksek  $R^2$  skoru ile maaş tahmini probleminde en iyi sonuçları sağlamıştır. Özellikle daha büyük veri kümelerinde ve kompleks model yapılarında, ReLU'nun lineer olmayan ancak pozitif doğrultuda sürekli olan yapısının modelin performansını artırdığı gözlemlenmiştir.

## KAYNAKÇA

Ataseven, Burçin. “Yapay sinir ağıları ile öngörü modellemesi.” Öneri Dergisi 10.39 (2013): 101-115.

Gökçay, M. Murat. “Yapay sinir ağıları ve uygulamaları.” MS thesis. Fen Bilimleri Enstitüsü, 1993.

Hamzaçebi, Coşkun. “Yapay Sinir Ağları – MATLAB Uygulamaları.” (2021)

Öztemel, Ercan. “Yapay sinir ağıları.” PapatyaYayincılık, İstanbul (2003).

Tosunoğlu, Nuray. “Zaman Serilerinin Öngörüsünde Yapay Sinir Ağları.” (2021)

Zhang, Zhihua, and Zhihua Zhang. "Artificial neural network." Multivariate time series analysis in climate and environmental research (2018): 1-35.

[1] [Yapay Sinir Ağı\(Artificial Neural Network\) Nedir? - Veri Bilimi Okulu - Veri Bilimi Okulu](#)

[2] [Yapay sinir ağıları - Vikipedi](#)

[3] [Yapay Sinir Ağları Nedir? | Yapay Zekâ | TÜBİTAK Bilim Genç](#)

[4] <https://miuul.com/blog/yapay-sinir-aglari>

[5] <https://www.geeksforgeeks.org/artificial-neural-networks-and-its-applications/>

[6] <https://acikders.ankara.edu.tr/course/view.php?id=6402>

[7] <https://dergipark.org.tr/tr/pub/ejosat/issue/52599/680773>