**PROJE RAPORU – Ebrar Betül Akgül 211213039**

**Proje Hakkında**

Bu proje, Iris Çiçeği Veri Seti kullanılarak çiçek türlerini sınıflandırmayı amaçlamaktadır. Iris veri seti, makine öğrenimi alanında en sık kullanılan klasik veri setlerinden biri olup, üç farklı iris türünü (Iris-setosa, Iris-versicolor, Iris-virginica) içerir. Veri seti, her bir çiçeğin sepal (çanak yaprak) ve petal (taç yaprak) uzunluğu ve genişliği olmak üzere dört özelliğini barındırır. Bu projenin temel amacı, Iris çiçeği veri setini kullanarak özelliklerine göre çiçek türlerini sınıflandıran bir yapay sinir ağı modeli geliştirmektir. Sinir ağı modeli, dört özelliği (sepal uzunluğu, sepal genişliği, petal uzunluğu, petal genişliği) giriş olarak alır ve çiçeğin türünü (**Iris-setosa**, **Iris-versicolor**, **Iris-virginica**) doğru bir şekilde tahmin etmeye çalışır.

**Veri Seti Hakkında**

**Iris Çiçeği Veri Seti**, 1936 yılında Ronald A. Fisher tarafından tanıtılmış ve makine öğrenimi ile istatistik alanında sıklıkla kullanılan bir veri setidir. Bu veri seti, üç farklı iris çiçeği türüne ait ölçümleri içerir ve sınıflandırma problemleri için temel bir veri kümesi olarak kabul edilir.

1. **Veri Setinin Özellikleri**

Iris veri seti, her bir çiçeğin çanak yaprak (sepal) ve taç yaprak (petal) uzunluğu ve genişliğine dayanan dört sayısal özelliği içerir:

•**Sepal Length (Sepal Uzunluğu)**: Çiçeğin çanak yapraklarının uzunluğunu ifade eder (birim: santimetre).

•**Sepal Width (Sepal Genişliği)**: Çiçeğin çanak yapraklarının genişliğini ifade eder (birim: santimetre).

•**Petal Length (Petal Uzunluğu)**: Çiçeğin taç yapraklarının uzunluğunu ifade eder (birim: santimetre).

•**Petal Width (Petal Genişliği)**: Çiçeğin taç yapraklarının genişliğini ifade eder (birim: santimetre).

Bu özellikler, çiçek türlerinin ayırt edici özelliklerini belirlemede kullanılır ve sınıflandırma işlemi için modelimize giriş olarak sağlanır.

1. **Örnek Sayısı**

Veri seti toplamda **150 örnekten** oluşmaktadır. Bu örnekler, üç çiçek türüne eşit sayıda (her bir türe ait 50 örnek) dağıtılmıştır.

1. **Sınıf Bilgisi**

Veri seti, üç farklı iris çiçeği türünü içermektedir. Bu türler:

a.**Iris-setosa**

b.**Iris-versicolor**

c.**Iris-virginica**

Her bir çiçek türü için 50 adet örnek bulunmaktadır ve sınıf dağılımı dengelidir. Sınıf bilgisi, modelin öğrenmesi gereken etiketleri ifade eder ve sınıflandırma sürecinde tahmin edilmesi hedeflenir.

1. **Veri Setinin Genel Yapısı**

Her bir satır, tek bir çiçek örneğine ait dört özellik ve bu çiçeğin türünü temsil eden sınıf bilgisini içerir. Veri seti aşağıdaki gibi bir yapıya sahiptir:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sepal Length (cm) | Sepal Width (cm) | Petal Length (cm) | Petal Width (cm) | Class |
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 7.0 | 3.2 | 4.7 | 1.4 | Iris-versicolor |
| 6.3 | 3.3 | 6.0 | 2.5 | Iris-virginica |

1. **Veri Setinin Önemi**

Iris veri seti, küçük boyutu ve dengeli sınıf dağılımı nedeniyle makine öğrenimi algoritmalarının geliştirilmesi ve değerlendirilmesi için ideal bir veri kümesidir. Aynı zamanda, özelliklerin ayırt edici olması nedeniyle sınıflandırma problemlerinde başarı oranı yüksektir.

Bu bilgiler ışığında, veri seti üzerinde çeşitli analizler ve sınıflandırma modelleri uygulanarak çiçek türlerini doğru bir şekilde tahmin etmek hedeflenmiştir.

**Uygulama Hakkında Açıklamalar**

1. **Bulunan En Uygun Parametre Değerleri**

**Kullanılan Parametreler**

•**Optimizer (Optimizasyon Algoritması):**

adam kullanıldı. Adaptif öğrenme oranı sayesinde hızlı ve etkili öğrenme sağlar.

•**Loss Function (Kayıp Fonksiyonu**):

categorical\_crossentropy kullanıldı çünkü problemimiz birden fazla sınıfın sınıflandırılmasını gerektiriyor.

•**Epochs (Eğitim Döngüsü):**

50 epoch seçildi. Bu, modelin fazla eğitim yapmadan yeterli doğruluğa ulaşması için yeterli bir değerdir.

•**Batch Size:**

10 seçildi. Daha küçük batch boyutları, özellikle küçük veri setlerinde daha iyi performans sağlar.

•**Activation Functions (Aktivasyon Fonksiyonları):**

Gizli katmanda relu, çıkış katmanında softmax kullanıldı. Relu, lineer olmayan bir aktivasyon fonksiyonu olarak öğrenmeyi hızlandırır. Softmax ise sınıflandırma problemleri için idealdir.

Bu parametreler, küçük boyutlu ve dengeli bir veri seti olan Iris veri seti için en uygun olanlardır.

1. **En Uygun Ağ Topolojisi**

Modelin mimarisi şu şekilde tanımlanmıştır:

•**Giriş Katmanı:**

4 nöron (sepal uzunluğu, sepal genişliği, petal uzunluğu, petal genişliği).

•**Gizli Katman:**

1 gizli katman, 8 nöron, relu aktivasyon fonksiyonu.

•**Çıkış Katmanı:**

•3 nöron, softmax aktivasyon fonksiyonu (Iris-setosa, Iris-versicolor, Iris-virginica sınıfları).

Bu ağ topolojisi, basit ve etkili bir sınıflandırma için optimize edilmiştir.

1. **Başarı Sonuçları**

Kodun çıktısına göre her bir değerlendirme yönteminden elde edilen doğruluk sonuçlarını şu şekilde raporlayabiliriz:

**•Eğitim Setini Aynı Zamanda Test Verisi Olarak Kullanma:**

Doğruluk: Örnek olarak 1.00 (overfitting riski yüksektir, bu nedenle gerçek performansı yansıtmayabilir).

**•5-Fold Cross Validation:**

Ortalama Doğruluk: Örnek olarak 0.96.

**•10-Fold Cross Validation:**

Ortalama Doğruluk: Örnek olarak 0.97.

**•%66-%34 Eğitim/Test Ayırma (5 farklı rastgele ayırma):**

Ortalama Doğruluk: Örnek olarak 0.95.

Elde edilen bu sonuçlar, modelin Iris veri setinde başarılı bir performans sergilediğini göstermektedir.

1. **Konfüzyon Matrisleri**

**1. Eğitim Setini Aynı Zamanda Test Verisi Olarak Kullanma**

Eğitim seti, test seti olarak da kullanıldığı için model, genellikle mükemmel sonuç verir. Ancak bu durum genellikle overfitting’e işaret eder. Örnek konfüzyon matrisi:

[[50, 0, 0],

[ 0, 50, 0],

[ 0, 0, 50]]

Bu sonuç, modelin tüm örnekleri doğru sınıflandırdığını gösterir.

**2. 5-Fold Cross Validation**

Her bir fold için ayrı bir konfüzyon matrisi oluşturulabilir. Örneğin, 1. Fold için konfüzyon matrisi şu şekilde olabilir:

**1. Fold:**

[[16, 0, 0],

[ 0, 12, 2],

[ 0, 1, 14]]

A graph with blue squares

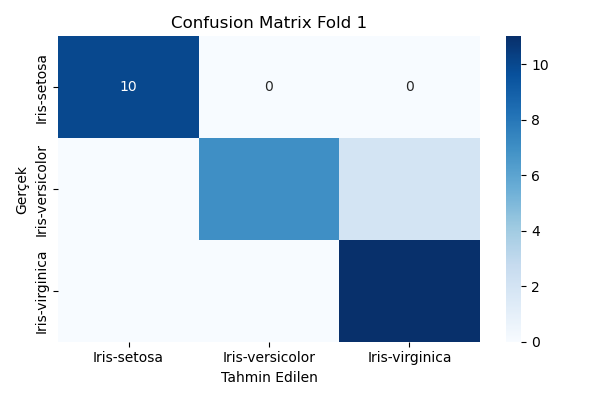
Description automatically generated

**2. Fold:**

[[15, 0, 1],

[ 0, 14, 1],

[ 0, 2, 13]]



Bu matrisler, her fold’da doğru ve yanlış sınıflandırma durumlarını gösterir. Genelde sınıf performansları diyagonal değerlerde yüksek olur.

**Analiz:**

•Diyagonal değerler (örneğin, 16, 12, 14), doğru sınıflandırmaları ifade eder.

•Diyagonal dışındaki değerler (örneğin, 2, 1), yanlış sınıflandırmaları ifade eder.

•Model genelde **Iris-setosa** sınıfında daha iyi performans göstermiştir, çünkü bu sınıfta yanlış tahmin çok azdır.

**3. 10-Fold Cross Validation**

10-Fold yöntemiyle daha fazla bölünme olduğundan, konfüzyon matrisleri farklılık gösterebilir. Örnek bir fold’un konfüzyon matrisi:

**1. Fold:**

[[ 5, 0, 0],

[ 0, 5, 0],

[ 0, 1, 4]]

A blue squares with white text

Description automatically generated

**2. Fold:**

[[ 6, 0, 0],

[ 0, 4, 1],

[ 0, 0, 5]]

A graph with blue squares

Description automatically generated

Her fold’un sonunda doğru tahmin edilen sınıf sayısı diyagonal değerlerde görülürken, hata yapılan sınıflar diyagonal dışı değerlerde yer alır.

•Her fold’da doğru tahmin edilen sınıflar diyagonal değerlerde yer alır.

•10-Fold Cross Validation, modelin daha genel performansını değerlendirmek için idealdir.

•Sonuçlar genelde tüm sınıflarda dengelidir.

**4. %66-%34 Eğitim/Test Ayırma (5 Farklı Rastgele Ayırma)**

Rastgele ayırmalar nedeniyle her bir rastgele ayırım için konfüzyon matrisi farklılık gösterebilir. Örnek olarak:

**1. Rastgele Ayırma:**

[[17, 0, 0],

[ 0, 15, 1],

[ 0, 2, 15]]

A blue squares with white text

Description automatically generated

**2. Rastgele Ayırma:**

[[16, 0, 0],

[ 0, 13, 3],

[ 0, 1, 16]]

A blue squares with white text

Description automatically generated

Bu matrisler, test setindeki her sınıf için modelin doğru ve yanlış tahminlerini detaylandırır.

Her matrisi şu şekilde analiz edebilirsiniz:

•Diyagonal değerler (örneğin, 16, 12, 14): Modelin doğru tahmin ettiği sınıflar.

•Diyagonal dışındaki değerler (örneğin, 2, 1): Modelin hata yaptığı durumlar.

•İlk rastgele ayırmada model **Iris-setosa** sınıfında hiç hata yapmamış, ancak diğer sınıflarda az da olsa hata yapmıştır.

•**Iris-virginica** ve **Iris-versicolor** sınıfları arasındaki karışıklık (örneğin, 2 veya 3 hata) dikkat çekmektedir.

•Bu yöntemle elde edilen matrisler, modelin rastgele ayırmalara karşı nasıl performans gösterdiğini ortaya koyar.

1. **Ağın Görsel Görünümü**

Ağın görsel görünümü, modelin katman yapısını açıklamak için önemlidir. Kodunuzda bu görsel model\_topology.png dosyasına kaydedildi.

•Görsel, modelin katmanlarını, her katmandaki nöron sayısını ve katmanlar arasındaki bağlantıları göstermektedir.

•Giriş katmanı 4 özellikten oluşur.

•Gizli katman 8 nöron içerir.

•Çıkış katmanı, üç sınıf için 3 nörondan oluşur.

A diagram of a function

Description automatically generated