

Yapay Zeka Projesi: Kategorileri Sınıflandırma

Hazırlayan: Tarık Kılıç Alsancak

Projenin Amacı

- Bu proje, belirli bir veri kümesindeki kategorileri doğru bir şekilde sınıflandırmak için yapay zeka ve derin öğrenme tekniklerini kullanmayı hedeflemektedir.

Kullanılan Yöntemler

- • Veri Ön İşleme: Verilerin temizlenmesi, normalizasyonu ve uygun formatta hazırlanması.
- • Eğitim: Modelin eğitiminde keras ve TensorFlow kütüphaneleri kullanıldı.
- • Değerlendirme: Eğitim ve doğrulama setleri üzerinde modelin performansının ölçülmesi.

Modeller

- 1- KNN
- 2- Random Forest
- 3- Decision Tree Classifier and Regressor
- 4- CNN model 1
- 5- CNN model 2

Deney Sonuçları

- KNN:

accuracy_score 0.8554

f1_score 0.8546439722018904

recall_score 0.8554

precision_score 0.8578152450755355

Random Forest

- `RandomForestClassifier(n_estimators=100,
random_state=42)`

`accuracy_score 0.8554`

`f1_score 0.8546439722018904`

`recall_score 0.8554`

`precision_score 0.8578152450755355`

Karar Ağacı

- DecisionTreeClassifier
- accuracy_score 0.7888
- f1_score 0.7896654287555152
- recall_score 0.7888
- precision_score 0.7907944419130593

Karar Ağacı

- DecisionTreeRegressor
- accuracy_score 0.7825
- f1_score 0.7828640198338868
- recall_score 0.7825
- precision_score 0.7834635976607696

CNN 1

- `model = models.Sequential([`
- `layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28,`
`1)),`
- `layers.MaxPooling2D((2, 2)),`
- `layers.Conv2D(64, (3, 3), activation='relu'),`
- `layers.MaxPooling2D((2, 2)),`
- `layers.Conv2D(64, (3, 3), activation='relu'),`
- `layers.Flatten(),`
- `layers.Dense(64, activation='relu'),`
- `layers.Dense(10, activation='softmax')`
- `])`

CNN 1

- Test doğruluğu: 0.90

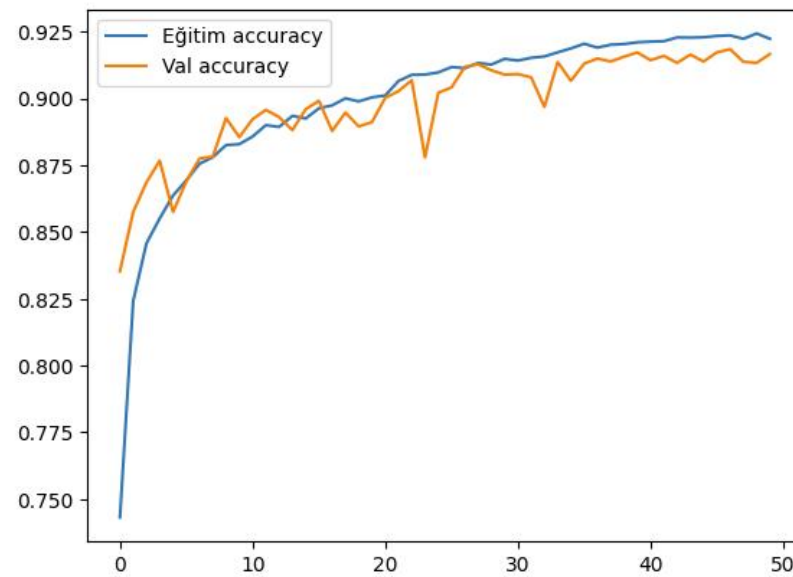
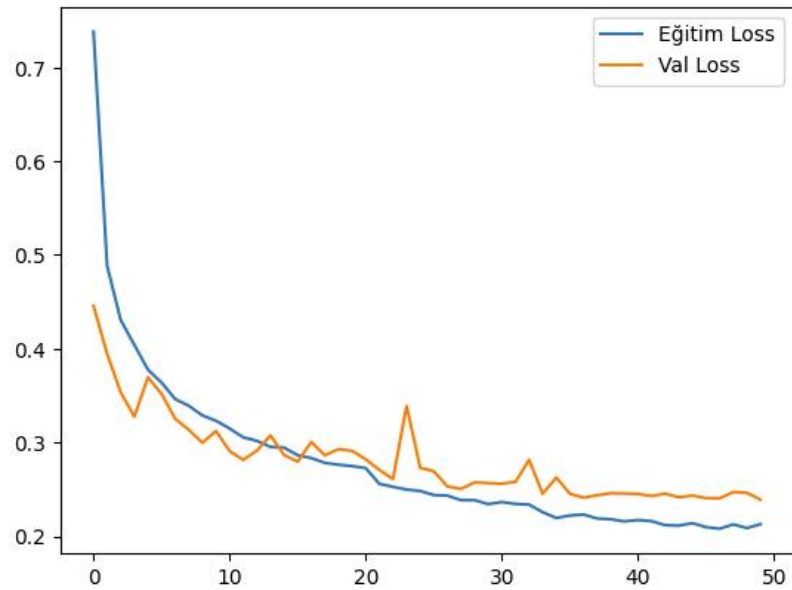
```
array([[862,  0, 11, 16,  3,  1, 101,  0,  6,  0],
       [  2, 966,  0, 25,  1,  0,  4,  0,  2,  0],
       [ 15,  1, 864,  5, 25,  0, 86,  0,  4,  0],
       [ 17,  1, 12, 900, 19,  0, 45,  0,  6,  0],
       [  0,  1, 75, 17, 798,  0, 97,  0, 12,  0],
       [  0,  0,  0,  1,  0, 983,  0, 11,  0,  5],
       [120,  0, 46, 23, 33,  1, 765,  0, 12,  0],
       [  0,  0,  0,  0,  0, 15,  0, 974,  0, 11],
       [  3,  1,  3,  2,  1,  2,  0,  4, 984,  0],
       [  1,  0,  0,  0,  0,  8,  0, 45,  0, 946]])
```

CNN 2

- `model = models.Sequential([`
- `# İlk Konvolüsyonel Katman`
- `layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),`
- `layers.BatchNormalization(),`
- `layers.MaxPooling2D((2, 2)),`
- `layers.Dropout(0.25),`
- `layers.Conv2D(64, (3, 3), activation='relu'),`
- `layers.BatchNormalization(),`
- `layers.MaxPooling2D((2, 2)),`
- `layers.Dropout(0.25),`
- `# Üçüncü Konvolüsyonel Katman`
- `layers.Conv2D(128, (3, 3), activation='relu'),`
- `layers.BatchNormalization(),`
- `layers.MaxPooling2D((2, 2)),`
- `layers.Dropout(0.25),`
- `# Tam Bağlantılı Katmanlar`
- `layers.Flatten(),`
- `layers.Dense(128, activation='relu'),`
- `layers.BatchNormalization(),`
- `layers.Dropout(0.5),`
- `# Çıkış Katmanı`
- `layers.Dense(10, activation='softmax')`
- `])`

CNN 2

loss: 0.2128 - accuracy: 0.9223 - val_loss: 0.2390 - val_accuracy: 0.9166 - lr: 1.2500e-04



CNN 2

```
array([[845, 1, 27, 12, 2, 1, 106, 0, 6, 0],
       [ 0, 977, 2, 14, 1, 0, 4, 0, 2, 0],
       [12, 1, 897, 10, 33, 0, 47, 0, 0, 0],
       [ 6, 5, 17, 914, 24, 0, 34, 0, 0, 0],
       [ 1, 0, 38, 21, 871, 0, 67, 0, 2, 0],
       [ 0, 0, 0, 2, 0, 977, 0, 10, 0, 11],
       [77, 0, 60, 23, 63, 0, 771, 0, 6, 0],
       [ 0, 0, 0, 0, 0, 11, 0, 965, 0, 24],
       [ 0, 0, 3, 3, 1, 2, 4, 0, 986, 1],
       [ 0, 0, 0, 0, 0, 3, 0, 33, 1, 963]])
```

Tartışma

- • Elimizdeki veri setine göre CNN modeller daha iyi çalışmaktadır
- • Tasarım parametrelerini değiştirerek daha iyi sonuçlar almak mümkündür.

Sonuç ve Gelecek Çalışmalar

MODEL	accuracy_score	f1_score	recall_score	precision_score
KNN	0.8554	0.8546439722	0.8554	0.8578152450
RANDOM FOREST	0.8554	0.8546439722	0.8554	0.8578152450
DecisionTreeClas sifier	0.7888	0.7896654287	0.7888	0.7907944419
DecisionTreeReg ressor	0.7825	0.7828640198	0.7825	0.7834635976
CNN 1	0.9042			
CNN 2	0.9166			