

# YÜKSEK DÜZEY PROGRAMLAMA DERSİ

## ENVER KANGAL/ 202213172049 / Normal Öğretim

### ÖDEV:

### Digit Recognizer:

Digit Recognizer veri seti, el yazısı rakamları sınıflandırmayı amaçlayan bir problemidir. Bu tür bir problem için Convolutional Neural Network (CNN) kullandım çünkü CNN'ler, özellikle görüntü işleme ve sınıflandırma gibi görsel veriler üzerinde oldukça başarılı sonuçlar veren bir mimardır.

#### Kullanım sebepleri:

1. Yerel öznitelik algılama yeteneği.
2. Uzaysal hiyerarşi oluşturma.
3. Parametre paylaşımı ile verimlilik.
4. Boyut azaltma (Pooling işlemleri).
5. Çevirme ve kaydırma dayanıklılığı.
6. Otomatik öznitelik çıkarma.
7. Görsel verilere uygunluk ve yüksek boyutlu verilerle başa çıkma.

### Proje kodları ve açıklamaları :

```
[15]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.utils import to_categorical
print("Tüm kütüphaneler başarıyla yüklendi!")

Tüm kütüphaneler başarıyla yüklendi!

[17]: !pip install tensorflow

Requirement already satisfied: tensorflow in /Users/enver1/anaconda3/lib/python3.11/site-packages (2.18.0)
Requirement already satisfied: absl-py>=1.0.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (2.0.0)
Requirement already satisfied: astunparse>=1.6.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (24.3.25)
Requirement already satisfied: gast!=0.5.0,!0.5.1,!0.5.2,>=0.2.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!4.21.1,!4.21.2,!4.21.3,!4.21.4,!4.21.5,<6.0.0dev,>=3.20.3 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (3.20.3)
Requirement already satisfied: requests<3,>=2.21.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (68.0.0)
Requirement already satisfied: six>=1.12.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt>=1.11.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (1.68.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (3.7.0)
Requirement already satisfied: numpy<2.1.0,>=1.26.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (1.26.4)
Requirement already satisfied: h5py>=3.11.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (3.11.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.33.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorflow) (0.33.1)
```

Gerekli kütüphaneler (ör. TensorFlow, NumPy, Pandas) projeye dahil edilip, TensorFlow'un sistemde kurulu olduğu doğrulandı.Kütüphaneler yüklendi mesajı verildi.TensorFlow yüklü değildi.

```
w) (0.37.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from astunparse==1.6.0->tensorflow) (0.38.4)
Requirement already satisfied: rich in /Users/enver1/anaconda3/lib/python3.11/site-packages (from keras>=3.5.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in /Users/enver1/anaconda3/lib/python3.11/site-packages (from keras>=3.5.0->tensorflow) (0.0.8)
Requirement already satisfied: optree in /Users/enver1/anaconda3/lib/python3.11/site-packages (from keras>=3.5.0->tensorflow) (0.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from requests<3,>=2.21.0->tensorflow) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from requests<3,>=2.21.0->tensorflow) (2024.7.4)
Requirement already satisfied: markdown>=2.6.8 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorboard<2.19,>=2.18->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from tensorboard<2.19,>=2.18->tensorflow) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from werkzeug>=1.0.1->tensorboard<2.19,>=2.18->tensorflow) (2.1.3)
Requirement already satisfied: markdown-it-py>=2.2.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from rich->keras>=3.5.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from rich->keras>=3.5.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~0.1 in /Users/enver1/anaconda3/lib/python3.11/site-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.0)

[notice] A new release of pip is available: 24.2 -> 24.3.1
[notice] To update, run: pip install --upgrade pip
```

```
[19]: import tensorflow as tf
print(tf.__version__) # TensorFlow sürümünü yazdırır

2.18.0
```

**TensorFlow Başarılı bir şekilde yüklendi ve komut ile versiyonunu kontrol ettik.**

```
[8]: import pandas as pd

# CSV dosyasını yükle
data = pd.read_csv("train.csv") # Dosya adını uygun şekilde değiştir

# İlk birkaç satırı gör
print(data.head())

# Veri setinin genel bilgilerini kontrol et
print(data.info())

  label  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6  pixel7  \
0      1      0      0      0      0      0      0      0      0
1      0      0      0      0      0      0      0      0      0
2      1      0      0      0      0      0      0      0      0
3      4      0      0      0      0      0      0      0      0
4      0      0      0      0      0      0      0      0      0

  pixel18  ...  pixel774  pixel775  pixel776  pixel777  pixel778  pixel779  \
0      0  ...      0      0      0      0      0      0
1      0  ...      0      0      0      0      0      0
2      0  ...      0      0      0      0      0      0
3      0  ...      0      0      0      0      0      0
4      0  ...      0      0      0      0      0      0

  pixel780  pixel781  pixel782  pixel783
0      0      0      0      0
1      0      0      0      0
2      0      0      0      0
3      0      0      0      0
4      0      0      0      0

[5 rows x 785 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 42000 entries, 0 to 41999
Columns: 785 entries, label to pixel783
dtypes: int64(785)
memory usage: 251.5 MB
None
```

CSV dosyası yüklendi, ilk 5 satır ve veri seti hakkında sütun, veri tipi, giriş sayısı ve bellek bilgileri görüntülendi.

```
[57]: import pandas as pd

# CSV dosyasını yükle
data = pd.read_csv("test.csv") # Dosya adını uygun şekilde değiştir

# İlk birkaç satırı gör
print(data.head())

# Veri setinin genel bilgilerini kontrol et
print(data.info())
```

	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	\
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	

	pixel9	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	\
0	0	...	0	0	0	0	0	0	
1	0	...	0	0	0	0	0	0	
2	0	...	0	0	0	0	0	0	
3	0	...	0	0	0	0	0	0	
4	0	...	0	0	0	0	0	0	

	pixel780	pixel781	pixel782	pixel783
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0

```

[5 rows x 784 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28000 entries, 0 to 27999
Columns: 784 entries, pixel0 to pixel783
dtypes: int64(784)
memory usage: 167.5 MB
None

```

test.csv dosyası yüklendi, ilk 5 satır ve veri setinin genel bilgileri (satır, sütun sayısı, veri türleri, bellek kullanımı) incelendi.

```
[21]: from tensorflow.keras.utils import to_categorical

# Etiketleri kategorik hale getir
y = to_categorical(y, num_classes=10)

print(f"Veri şekli: {X.shape}")
print(f"Etiket şekli: {y.shape}")

Veri şekli: (42000, 28, 28, 1)
Etiket şekli: (42000, 10)
```

```
[23]: from sklearn.model_selection import train_test_split

# Veriyi eğitim ve test setine böl
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f"Eğitim verisi şekli: {X_train.shape}")
print(f"Test verisi şekli: {X_test.shape}")

Eğitim verisi şekli: (33600, 28, 28, 1)
Test verisi şekli: (8400, 28, 28, 1)
```

```
[27]: from tensorflow.keras.layers import Input

model = Sequential([
    Input(shape=(28, 28, 1)), # İlk katmanda Input nesnesi kullanıyoruz
    Conv2D(32, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(10, activation='softmax')
])
```

```
[29]: model.compile(optimizer='adam',
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])
```

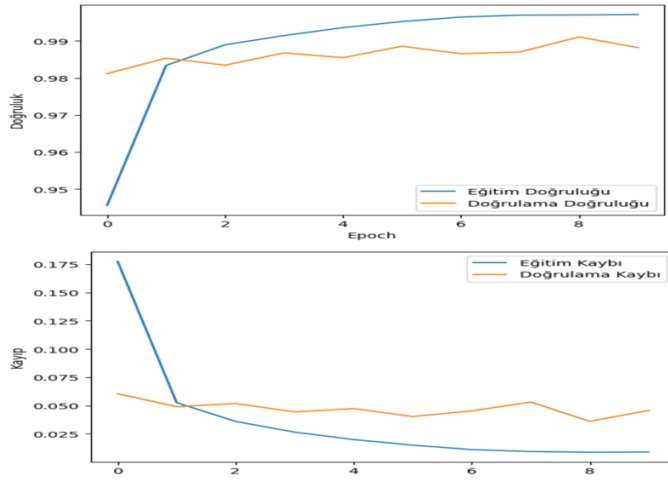
Eğitim ve test verileri hazırlandı, bir CNN modeli tanımlandı ve derlendi.

```
[31]: history = model.fit(X_train, y_train,
                        epochs=10, # 10 epoch boyunca eđit
                        validation_data=(X_test, y_test),
                        batch_size=32) # Her iterasyonda 32 örnek kullan

Epoch 1/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.8771 - loss: 0.4018 - val_accuracy: 0.9812 - val_loss: 0.0604
Epoch 2/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9817 - loss: 0.0550 - val_accuracy: 0.9854 - val_loss: 0.0490
Epoch 3/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9896 - loss: 0.0335 - val_accuracy: 0.9835 - val_loss: 0.0518
Epoch 4/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9909 - loss: 0.0275 - val_accuracy: 0.9868 - val_loss: 0.0443
Epoch 5/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9953 - loss: 0.0163 - val_accuracy: 0.9855 - val_loss: 0.0473
Epoch 6/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9957 - loss: 0.0143 - val_accuracy: 0.9886 - val_loss: 0.0403
Epoch 7/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9965 - loss: 0.0104 - val_accuracy: 0.9865 - val_loss: 0.0452
Epoch 8/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9974 - loss: 0.0083 - val_accuracy: 0.9870 - val_loss: 0.0531
Epoch 9/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9972 - loss: 0.0075 - val_accuracy: 0.9911 - val_loss: 0.0360
Epoch 10/10
1050/1050 ————— 6s 6ms/step - accuracy: 0.9973 - loss: 0.0086 - val_accuracy: 0.9882 - val_loss: 0.0456

[33]: test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Test dođruluk oranı: {test_acc:.2f}")
263/263 ————— 0s 2ms/step - accuracy: 0.9885 - loss: 0.0521
Test dođruluk oranı: 0.99
```

Model eđitildi, test verisiyle deđerlendirildi ve performansı gorselleřtirildi.



```
import matplotlib.pyplot as plt

# Eđitim ve dođrulama dođruluk oranları
plt.plot(history.history['accuracy'], label='Eđitim Dođruluđu')
plt.plot(history.history['val_accuracy'], label='Dođrulama Dođruluđu')
plt.xlabel('Epoch')
plt.ylabel('Dođruluk')
plt.legend()
plt.show()

# Eđitim ve dođrulama kayıpları
plt.plot(history.history['loss'], label='Eđitim Kaybı')
plt.plot(history.history['val_loss'], label='Dođrulama Kaybı')
plt.xlabel('Epoch')
plt.ylabel('Kayıp')
plt.legend()
plt.show()
```

Eđitim ve dođrulama süreçlerinin dođruluk ve kayıp deđerleri grafikte gorselleřtirildi

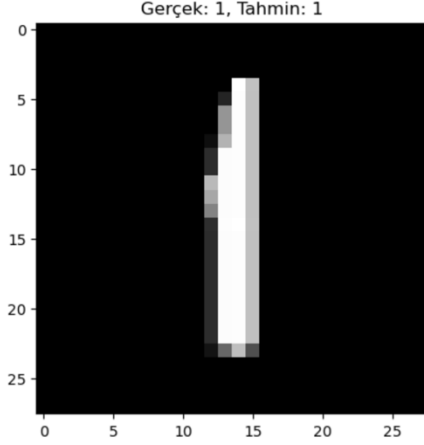
```
[43]: model.save("digit_recognizer_model.keras")
```

```
[45]: import numpy as np
```

```
# Test seti üzerinde tahmin yap
predictions = model.predict(X_test)

# Rastgele bir örneği seç ve sonucu gör
index = np.random.randint(0, len(X_test))
plt.imshow(X_test[index].reshape(28, 28), cmap='gray')
plt.title(f"Gerçek: {np.argmax(y_test[index])}, Tahmin: {np.argmax(predictions[index])}")
plt.show()
```

263/263 — 0s 1ms/step



Bu projede, el yazısı rakamlarını tanımak için bir sinir ağı modeli eğiterek, modelin test verisi üzerindeki doğruluğunu değerlendirdim ve rastgele seçilen bir test örneğinde, modelin tahminini gerçek değerle karşılaştırarak görselleştirdim.

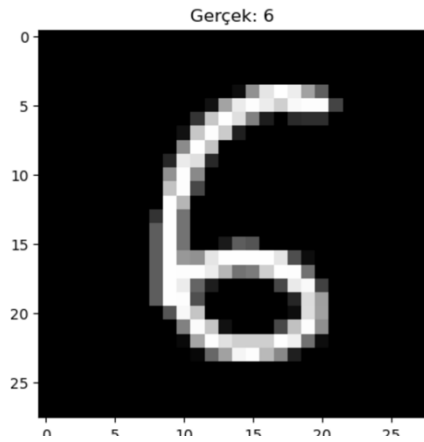
```
[51]: import numpy as np
```

```
# Test setinde etiketleri 7 olanları bul
six_indices = np.where(np.argmax(y_test, axis=1) == 6)[0]

# İlk 6 örneğini seç
index = six_indices[0] # 6 rakamına ait ilk örneğin indeksini al

# Görüntüyü çizdir
plt.imshow(X_test[index].reshape(28, 28), cmap='gray')
plt.title("Gerçek: 6")
plt.show()

# Tahmini yap
prediction = model.predict(X_test[index].reshape(1, 28, 28, 1)) # Modelin beklediği şekle göre yeniden boyutlandır
print(f"Modelin Tahmini: {np.argmax(prediction)}")
```



Bu koddaki, test setinde "6" rakamına ait bir örnek bulunarak görselleştirildi ve modelin bu örnek üzerindeki tahmini gerçek değerle karşılaştırıldı.

