

# Digit Recognizer

Okan Kiraz 202113171024 Normal Öğretim

2024-2025 Güz Dönemi Yüksek Düzey Programlama Proje Ödevi

## Giriş

Projemizde derin öğrenme modeli, el yazısıyla yazılmış rakamları tanımayı amaçlamaktadır. Model, TensorFlow ve Keras kütüphaneleri kullanılarak oluşturulmuş ve MNIST veri seti üzerinde eğitilmiştir. Bu veri seti, 0-9 arasındaki rakamların el yazısı ile yazılmış görüntülerini içermektedir. Model, verilen bir görüntüyü doğru bir şekilde rakam sınıfına ayırmayı öğrenmiş ve tahminler yapmıştır.

## Veri Seti

Bu projede kullanılan veri seti, MNIST (Modified National Institute of Standards and Technology) veri setidir. Bu veri seti, el yazısıyla yazılmış rakamların 60.000 eğitim ve 10.000 test örneğinden oluşmaktadır. Her bir görüntü 28x28 piksel boyutundadır ve bu veri seti, makine öğrenmesi için yaygın olarak kullanılan bir görüntü sınıflandırma veri setidir.

## Model Katmanları

Model, aşağıdaki katmanlardan oluşmaktadır:

- \*\*Flatten Katmanı\*\*:** 28x28 boyutundaki görüntüler 784 boyutlu bir vektöre düzleştirilir.
- \*\*Gizli Katman\*\*:** 128 nörondan oluşan ve ReLU aktivasyon fonksiyonu kullanan bir yoğun katman eklenmiştir.
- \*\*Çıkış Katmanı\*\*:** 10 nörondan oluşan ve Softmax aktivasyon fonksiyonu kullanan çıkış katmanı eklenmiştir. Bu katman, her rakam için bir olasılık değerini hesaplar.

## Modelin Eğitilmesi

Model, Adam optimizasyon algoritması ve sparse categorical cross-entropy kayıp fonksiyonu ile eğitilmiştir. Eğitim süreci 5 epoch gerçekleştirilmiştir. Bu süreçte model, her epoch sonunda ağırlıklarını (bir diğer deyişle öğrenilen parametrelerini) güncelleyerek veri setindeki el yazısı rakamları doğru bir şekilde sınıflandırmaya çalışmıştır.

## Sonuçlar

Modelin başarı oranı, test verisi üzerinde yapılan değerlendirmelerde %98 civarlarında elde edildi. Bu oldukça yüksek bir seviye başarı.

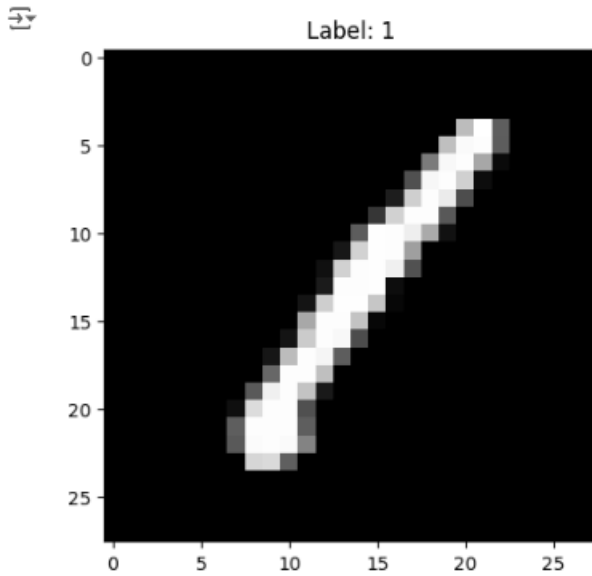
**Model eğitilmeden önce aşağıdaki kodun çıktısı olarak ilk eğitim örneğinin görseli ekranda gösterilecek ve etiketi başlık olarak yukarıda yazacaktır.**

```
✓ [18] # Eğitim verilerini hazırlıyoruz
1s x_train = train_data.drop('label', axis=1).values # 'label' sütununda işlem yapalım
y_train = train_data['label'].values

# Test verileri ayarları
x_test = test_data.values

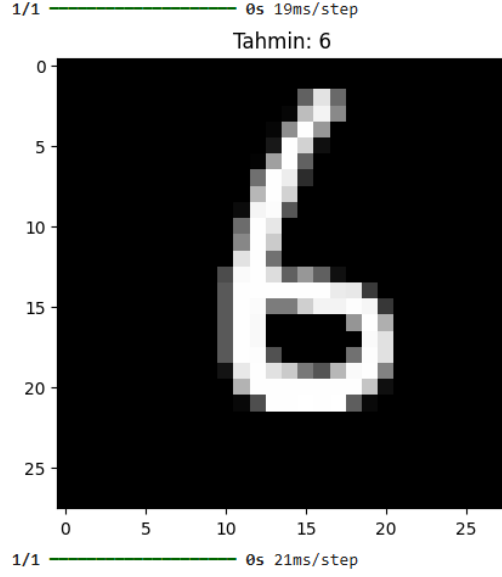
# Verileri normalleştirme
x_train = x_train / 255.0
x_test = x_test / 255.0

# Eğitim verisinin bir örneğini görselleştirelim
import matplotlib.pyplot as plt
plt.imshow(x_train[0].reshape(28, 28), cmap='gray') # 28x28 bir görsel ekrana gelsin
plt.title(f"Label: {y_train[0]}")
plt.show()
```

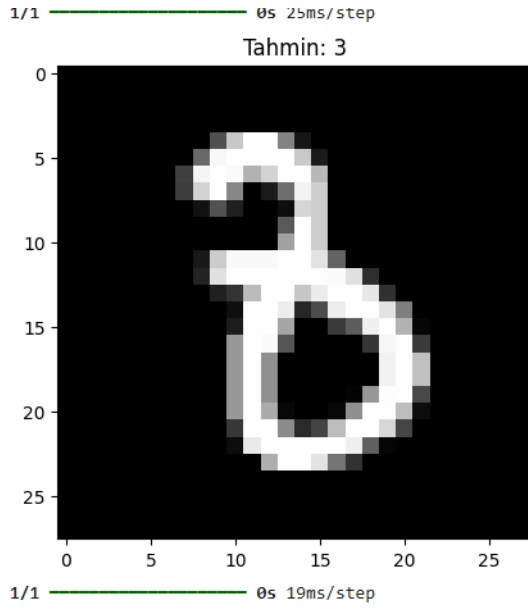


## Modelin Eğitilmesi Sonucu Oluşan Çıktılar (Kodda 5 Örnek Yazıldı)

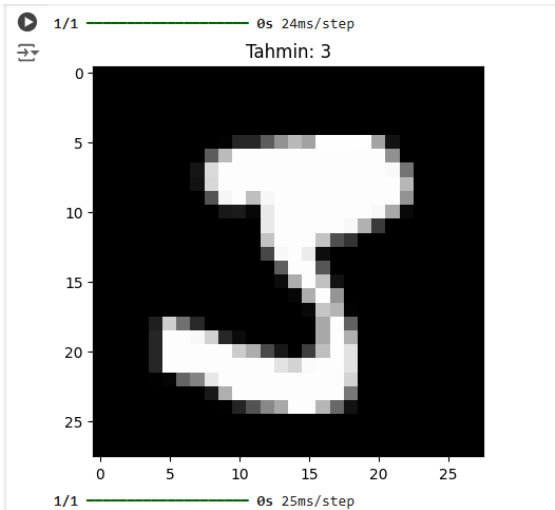
### Doğru Tahmin



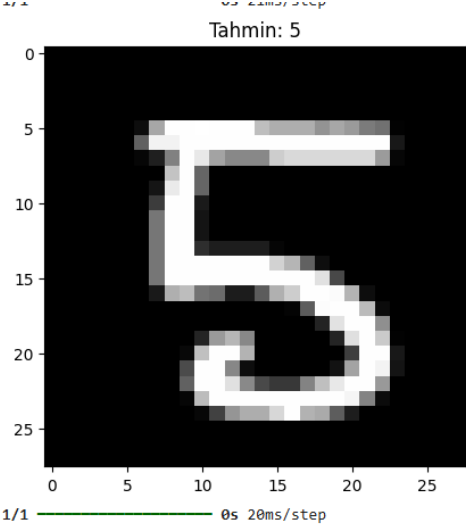
### Yanlış Tahmin



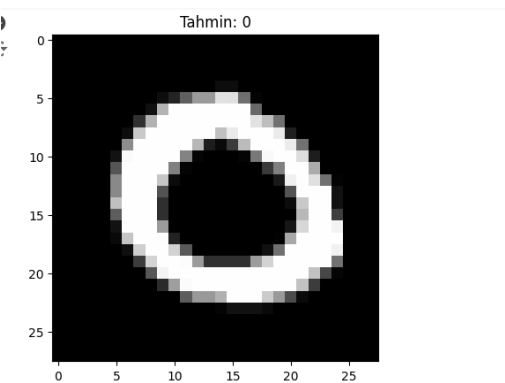
Doğru Tahmin



Doğru Tahmin



Doğru Tahmin



2. Bu dosyayı Drive'a kaydedtim. (Dosya Drive'a kaydedildi)

	A	B
1	ImageId	Label
2	1	2
3	2	0
4	3	9
5	4	9
6	5	3
7	6	7
8	7	0
9	8	3
10	9	0
11	10	3
12	11	5
13	12	7
14	13	4
15	14	0
16	15	4
17	16	3
18	17	3
19	18	1
20	19	9
21	20	0
22	21	9
23	22	1
24	23	1
25	24	5
26	25	7
27	26	4
28	27	2
29	28	7