



Derin Öğrenme (FET312) 2025-2026 Güz Dönemi

Dr. YILDIZ KARADAYI

**MLP ve CNN Karşılaştırmasıyla MNIST Veri Seti
Üzerinde El Yazısı Rakam Sınıflandırması**

DeepLeaders

- Apdullah Yuşa Önder 22040101029
apdullahyusaonder@stu.topkapi.edu.tr
- Hasan Önder Sert 21040301077
hasanondersert@stu.topkapi.edu.tr

GitHub/Repo bağlantısı:

[https://github.com/yusaonder/mnist_deeplearning]

İÇİNDEKİLER

1. Giriş ve Proje Özeti.....	2
2. Problem Tanımı & Motivasyon.....	2
3. Proje Yönetimi.....	3
4. İlgili Çalışmalar (Literatür Taraması)	3
○ 4.1 Temel Kaynaklar ve Kiyaslama.....	3
○ 4.2 Projemizin Farklılaşması.....	3
5. Veri Açıklaması ve Yönetimi.....	4
6. Yöntemler ve Mimari.....	4
○ 6.1 Veri Ön İşleme ve Hazırlık.....	4
○ 6.2 Yöntem 1: Çok Katmanlı Algılayıcı (MLP) Mimarisi.....	5
○ 6.3 Yöntem 2: Evrişimsel Sinir Ağrı (CNN) Mimarisi.....	6
7. Deney Tasarımı.....	7
8. Kullanılan Araçlar ve Frameworkler.....	7
9. Sonuçlar ve Bulgular.....	7
10. Gelecek Çalışmalar ve Planlanan Gelişmiş Mimariler.....	10
11. Kaynaklar.....	10

1) Giriş ve Proje Özeti

Bu proje görüntü sınıflandırma alanının veri kümesi olan MNIST El Yazısı Rakam Veri Seti üzerinde gerçekleştirilmiştir. Projenin ana hedefi, el yazısıyla yazılmış 0'dan 9'a kadar olan rakamların yüksek doğrulukla tanınması ve sınıflandırılmasıdır. Çalışmada, derin öğrenme mimarisinin görüntü verileri üzerindeki kritik rolünü deneyel olarak göstermek amacıyla iki farklı ağ türü geliştirilmiş ve karşılaştırılmıştır:

1. **Çok Katmanlı Algılayıcı (MLP):** Uzamsal bilgiyi kaybetmesi beklenen temel model
2. **Evrişimsel Sinir Ağrı (CNN):** Görüntüdeki yerel örüntüler (kenarlar, şekiller) yakalama ve konum değişmezliği özellikleriyle bilinen gelişmiş model.

Tüm modeller, PyTorch derin öğrenme kütüphanesi kullanılarak geliştirilmiş ve aynı eğitim parametreleri altında 10 epoch boyunca eğitilmiştir. Raporun ilerleyen bölümlerinde, her iki modelin mimari detayları, veri ön işleme adımları ve deneyel sonuçları sunulacaktır.

2) Problem Tanımı & Motivasyon

İş/Bilimsel Soru: El yazısıyla yazılmış 0-9 arasındaki rakamların bulunduğu görüntüler, derin öğrenme yöntemleri kullanarak doğru bir şekilde tanımak ve sınıflandırmaktır. Modelin, verilen bir görüntünün hangi rakam olduğunu otomatik olarak tahmin edebilmesi hedeflenmektedir. Bu problem, görüntü işleme ve örüntü tanıma alanında temel bir zorluk teşkil etmektedir.

Görev Türü: Sınıflandırma (10 sınıflı çoklu sınıflandırma).

Hedef Değişken(ler): Görüntüdeki el yazısı rakamı temsil eden tamsayı etiket (label) (0'dan 9'a kadar). Birimi yoktur.

Başarı Kriterleri: Nicel hedefler, kullanılan modelin Doğruluk (Accuracy) metriklerinin mümkün olduğunda yüksek olmasıdır. Öncelikli hedef, test/doğrulama veri setinde Doğruluk elde etmektir.

3) Proje Yönetimi

- 1. Hafta (20-27 Ekim): Veri seti ve proje konusunun seçimi. Sorumlular: DeepLeaders (Ortak)
- 2. Hafta (3-10 Kasım): Veri inceleme, keşifsel veri analizi (EDA) ve ön işleme adımlarının tamamlanması. Sorumlular: DeepLeaders (Ortak)
- 3-4. Hafta: Base Model Tasarımı ve Geliştirme (MLP ve CNN). İlk eğitim denemelerinin yapılması.
- 5-6. Hafta: Hiper Parametre Optimizasyonu. Modellerin performansının artırılmasına yönelik detaylı deneylerin yapılması.
- 7. Hafta: Performans Analizi ve Değerlendirme. Seçilen metriklere göre modellerin kıyaslanması ve final modelin seçimi.
- 8. Hafta: Proje Raporunun Son Hali ve Sunumun oluşturulması. Son çıktıların GitHub reposuna yüklenmesi.

Roller ve Sorumluluklar:

- Hasan Önder Sert (21040301077): MLP (Çok Katmanlı Algılayıcı) base modelinin geliştirilmesi ve değerlendirilmesi.
- Apdullah Yuşa Önder (22040101029): CNN (Evrişimsel Sinir Ağları) base modelinin geliştirilmesi ve değerlendirilmesi.
- Final Rapor ve Sunum: Ortak sorumluluk.

Çıktılar:

- Son Proje Raporu (PDF formatında)
- Hasan Önder Sert'in Kodu: [FET312_21040301077_DeepLeaders_1.ipynb](#) (MLP)
- Apdullah Yuşa Önder'in Kodu: [FET312_22040101029_DeepLeaders_1.ipynb](#) (CNN)
- Veri Seti dosyaları ([mnist_train.csv](#), [mnist_test.csv](#))

4) İlgili Çalışmalar (Literatür Taraması)

4.1 Temel Kaynaklar ve Kıyaslama

MNIST veri seti, görüntü sınıflandırımda bir benchmark (kıyaslama) olarak kabul edilir. Bu alandaki temel yaklaşımalar:

- LeNet-5 (Y. LeCun vd., 1998): MNIST üzerinde en başarılı erken mimaridir. Bu çalışma, Evrişimsel Sinir Ağlarının (CNN) görüntü verilerinde uzamsal bilgiyi koruma yeteneği sayesinde MLP'lere göre üstünlüğünü kanıtlamıştır. LeNet-5 varyantları ile %0.2'nin altında hata oranları raporlanmıştır.
- Çok Katmanlı Algılayıcı (MLP): Basit bir derin öğrenme tabanı olarak kullanılır. MLP'ler, piksel verilerini düzeltirerek uzamsal bilgiyi kaybettiği için, aynı problem üzerinde CNN'lere kıyasla daha düşük doğruluk (%97.0-98.0) ve daha yüksek aşırı uyum (overfitting) riski taşır.

4.2 Projemizin Farklılaşması

Projemiz, literatürdeki bu iki temel mimariyi, güncel bir derin öğrenme framework'ü olan PyTorch çatısı altında karşılaştırarak bir boşluğu doldurmaktadır.

- Karşıt Yaklaşım Kiyaslaması: Proje, görüntülerin uzamsal yapısını koruyan CNN (Apdullah Yuşa Önder) ile uzamsal bilgiyi düzleştiren MLP (Hasan Önder Sert) modellerini aynı veri kümesi ve aynı eğitim koşulları altında kıyaslamaktadır.
- Amaç: Bu pratik karşılaştırma ile, veri kümesi üzerinde bile CNN'in uzamsal özellik çıkarma yeteneğinin sağladığı performans avantajını nice sonuçlarla göstermeyi hedeflemektedir.

5) Veri Açıklaması ve Yönetimi

Veri Kümesi Açıklaması:

- Ad: MNIST (Modified National Institute of Standards and Technology) - CSV formatında.
- Kaynak/Bağlantı: <https://www.kaggle.com/datasets/oddrationale/mnist-in-csv>
- Lisans/Kullanım Hakları: Kaggle tarafından genel kullanımına sunulmuş bir veri kümesidir.

Şema:

- Öznitelikler: 784 adet piksel değeri (pixel0'dan pixel783'e). Tür: Tamsayı (0-255 gri tonlama).
- Hedef Değişken: `label`. Tür: Tamsayı (0-9).

Boyut:

- Görüntü Boyutu: 28 x28 piksel.
- Toplam Satır Sayısı: Eğitim verisi 60.000, Test verisi 10.000 olan toplam 70.000 görüntü.
- Kolon Sayısı: 785 (1 etiket + 784 piksel).
- Sınıf Dengesi: Veri seti, 0'dan 9'a kadar 10 sınıf arasında dengeli bir dağılıma sahiptir.

Etik, Gizlilik, Önyargı: Veri seti el yazısı rakamlarından oluşan kişisel veri veya hassas alanlar içermemektedir. Bu nedenle gizlilik veya etik endişeler minimum düzeydedir.

6) Yöntemler ve Mimari

Projemiz, el yazısı rakam tanıma problemini çözmek için PyTorch derin öğrenme kütüphanesi kullanılarak iki farklı temel sinir ağı mimarisi geliştirmiştir.

6.1 Veri Ön İşleme ve Hazırlık

Her iki model de aynı veri ön işleme adımlarını paylaşmıştır:

1. Veri Yükleme ve Ayırma: CSV formatındaki MNIST veri seti (`mnist_train.csv`), piksel verileri (öznitelikler) ve etiketler (hedef değişken) olarak ayrılmıştır.
2. Normalizasyon: 28x28 piksel boyutundaki gri tonlamalı görüntülerin piksel değerleri (0-255 aralığında) 255.0 değerine bölünerek 0.0 ile 1.0 aralığına ölçeklendirilmiştir (Normalizasyon). Bu, eğitim sürecinin hızlanması ve optimizasyonun iyileşmesi için kritik bir adımdır.

3. Veri Tipi Dönüşümü: NumPy dizileri, PyTorch'un `torch.Tensor` formatına dönüştürülmüştür. Etiketler (0-9), Kayıp Fonksiyonu (`nn.CrossEntropyLoss`) gereksinimlerine uygun olarak uzun tamsayı (`torch.long`) tipine dönüştürülmüştür.
4. Veri Yükleyicileri (DataLoaders): Veri setleri, toplu işleme (batching) ve karıştırma (shuffling) için `torch.utils.data.DataLoader` nesnelerine dönüştürülmüştür. Mini-Batch boyutu her iki model için de 64 olarak belirlenmiştir.

MLP MİMARİSİ İLE CNN MİMARİSİ KARŞILAŞTIRMASI

Özellik	Çok Katmanlı Algılayıcı (MLP)	Evrimsel Sinir Ağı (CNN)
Girdi Şekli	Düzleştirilmiş 1D Vektör (784 nöron)	3D Görüntü Tensörü ($1 \times 28 \times 28$)
Uzamsal Bilgi İşleme	Yok edilir (Komşuluk ilişkisi kullanılmaz).	Korunur (Yerel örüntülerin çıkarılır).
Temel Katman Tipi	Tam Bağlantılı (<code>nn.Linear</code>)	Evrşim (<code>nn.Conv2d</code>) ve Havuzlama
Parametre Paylaşımı	Yoktur.	Vardır (Ağırlıklar tüm girdi üzerinde paylaşılır).
MNIST Başarı Beklentisi	İyi (~%97.0–98.0)	Çok Yüksek (~%99.0+)

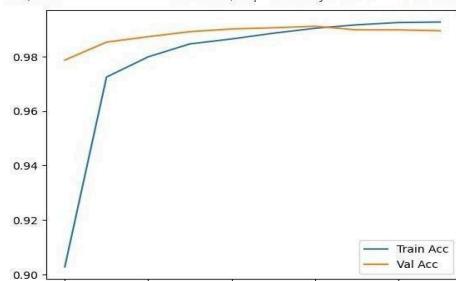
6.2 Yöntem 1: Çok Katmanlı Algılayıcı (MLP) Mimarisi

Bu model, Hasan Önder Sert tarafından geliştirilmiştir. Görüntülerin uzamsal bilgiyi dikkate almadan düzleştirilmiş bir vektör olarak işlenir.

```

Epoch 1/10
375/375      - 12s 26ms/step - accuracy: 0.9028 - loss: 0.3133 - val_accuracy: 0.9787 - val_loss: 0.0764
Epoch 2/10
375/375      - 9s 24ms/step - accuracy: 0.9724 - loss: 0.0902 - val_accuracy: 0.9852 - val_loss: 0.0505
Epoch 3/10
375/375      - 9s 24ms/step - accuracy: 0.9799 - loss: 0.0659 - val_accuracy: 0.9872 - val_loss: 0.0419
Epoch 4/10
375/375      - 9s 24ms/step - accuracy: 0.9846 - loss: 0.0509 - val_accuracy: 0.9891 - val_loss: 0.0378
Epoch 5/10
375/375      - 11s 29ms/step - accuracy: 0.9864 - loss: 0.0433 - val_accuracy: 0.9901 - val_loss: 0.0381
Epoch 6/10
375/375      - 12s 32ms/step - accuracy: 0.9885 - loss: 0.0368 - val_accuracy: 0.9905 - val_loss: 0.0324
Epoch 7/10
375/375      - 12s 32ms/step - accuracy: 0.9903 - loss: 0.0306 - val_accuracy: 0.9911 - val_loss: 0.0330
Epoch 8/10
375/375      - 21s 33ms/step - accuracy: 0.9916 - loss: 0.0276 - val_accuracy: 0.9898 - val_loss: 0.0333
Epoch 9/10
375/375      - 12s 32ms/step - accuracy: 0.9924 - loss: 0.0239 - val_accuracy: 0.9898 - val_loss: 0.0350
Epoch 10/10
375/375      - 12s 32ms/step - accuracy: 0.9926 - loss: 0.0231 - val_accuracy: 0.9894 - val_loss: 0.0356

```



Mimari Detayları

MLP modeli, 3 Tam Bağlantılı (Fully Connected) katmandan oluşmaktadır:

1. Giriş Katmanı (`nn.Flatten` ve `nn.Linear`):
 - o Girdi verisi, $28 \times 28 = 784$ 8 boyutunda tek bir vektöre dönüştürülür.
 - o İlk `nn.Linear` katmanı 784 girdiyi 256 nörona eşler.
 - o Aktivasyon: ReLU (`nn.ReLU`).
2. Gizli Katman:
 - o 256 girdiyi 128 nörona eşleyen inci bir `nn.Linear` katmanı kullanılır.
 - o Aktivasyon: ReLU (`nn.ReLU`).

3. Çıkış Katmanı:

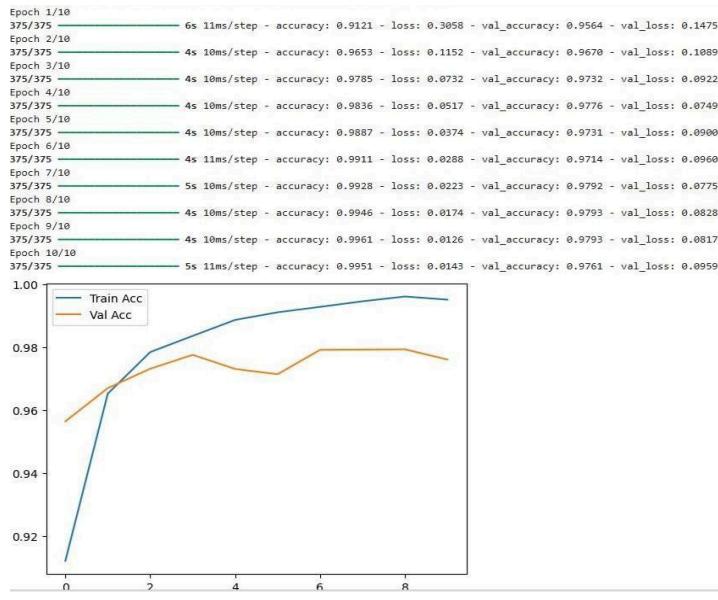
- o 128 girdiyi 0'dan 9'a kadar olan 10 sınıfı temsil eden 10 nörona eşleyen son `nn.Linear` katmanı.
- o Aktivasyon: Bu katmanda özel bir aktivasyon (örneğin Softmax) kullanılmamıştır, çünkü `nn.CrossEntropyLoss` kayıp fonksiyonu, Softmax işlevini bünyesinde barındırmaktadır.

Eğitim Parametreleri

- Optimizasyon: Adam (`torch.optim.Adam`)
- Öğrenme Oranı (Learning Rate): 0.001
- Kayıp Fonksiyonu (Loss Function): `nn.CrossEntropyLoss`
- Epoch Sayısı: 10

6.3 Yöntem 2: Evrişimsel Sinir Ağı (CNN) Mimarisi

Bu model, Apdullah Yuşa Önder tarafından geliştirilmiştir ve görüntülerin uzamsal yapısını koruyarak yerel özellikleri çıkarmayı hedefler.



Mimari Detayları

CNN modeli, 2 Evrişim bloğu ve 2 Tam Bağlantılı (Fully Connected) katmandan oluşmaktadır:

1. Veri Şekillendirme: Girdi tensörleri, 784 vektörden 1x28x28 sekline dönüştürülür.
2. Evrişim Bloğu 1:
 - `nn.Conv2d`: 1 girdi kanalı, 32 çıktı kanalı, 3x3 kernel boyutu.
 - `nn.ReLU`: Doğrusallık dışı aktivasyon.
 - `nn.MaxPool2d` 2x2 boyutuyla havuzlama (özellik haritası boyutunu yarıya indirir).
3. Evrişim Bloğu 2:
 - `nn.Conv2d`: 32 girdi kanalı, 64 çıktı kanalı, 3x3 kernel boyutu.
 - `nn.ReLU`: Doğrusallık dışı aktivasyon.
 - `nn.MaxPool2d`: 2x2 boyutuyla havuzlama.
4. Düzleştirme ve Tam Bağlantılı Katmanlar:
 - `nn.Flatten`: Evrişim katmanlarından çıkan 3 boyutlu özelliği (bu aşamada 64x5x5 boyutundadır.) tek bir vektöre dönüştürür.
 - `nn.Linear`: Vektör girdiyi 128 nörona eşler.
 - `nn.Dropout(0.4)`: Aşırı uyumu (overfitting) azaltmak için 0.4 oranıyla gizli nöronları devre dışı bırakır.
5. Çıkış Katmanı:
 - 128 girdiyi 10 sınıfına eşleyen son `nn.Linear` katmanı.

Eğitim Parametreleri

- Optimizasyon: Adam (`torch.optim.Adam`)
- Öğrenme Oranı (Learning Rate): 0.001
- Kayıp Fonksiyonu (Loss Function): `nn.CrossEntropyLoss`
- Epoch Sayısı: 10

7) Deney Tasarımı

Ana Amaç ve Yaklaşım

- Ana Amaç: Görüntü sınıflandırma görevinde, CNN mimarisinin uzamsal bilgiyi dikkate almayan MLP mimarisine göre performans üstünlüğünü nicel olarak doğrulamaktır
- Geliştirilen Modeller: İki temel model (MLP ve CNN) aynı koşullar altında PyTorch kullanılarak karşılaştırılmıştır

Veri Hazırlığı ve Validasyon

- Train/Validation Ayrımı: 60.000 eğitim görüntüsü, %80 Eğitim ve %20 Doğrulamaya ayrılmıştır
- Validasyon Yöntemi: Modelin genelleme yeteneğini izlemek ve aşırı uyumu tespit etmek için her epoch sonunda Doğrulama (Validation) seti üzerinde performans ölçümü yapılmıştır

Değerlendirme Kriterleri

- Temel Metrik: Sınıf dengesi nedeniyle ana kriter Doğruluk (Accuracy) olarak belirlenmiştir
- Başarı Hedefi: Doğruluk büyktür 0.98 hedeflenmektedir
- Sıralama Beklentisi: Uzamsal özellik çıkarma yeteneği nedeniyle CNN Modelinin, MLP modeline göre daha yüksek Doğruluk değerine ulaşması beklenmektedir

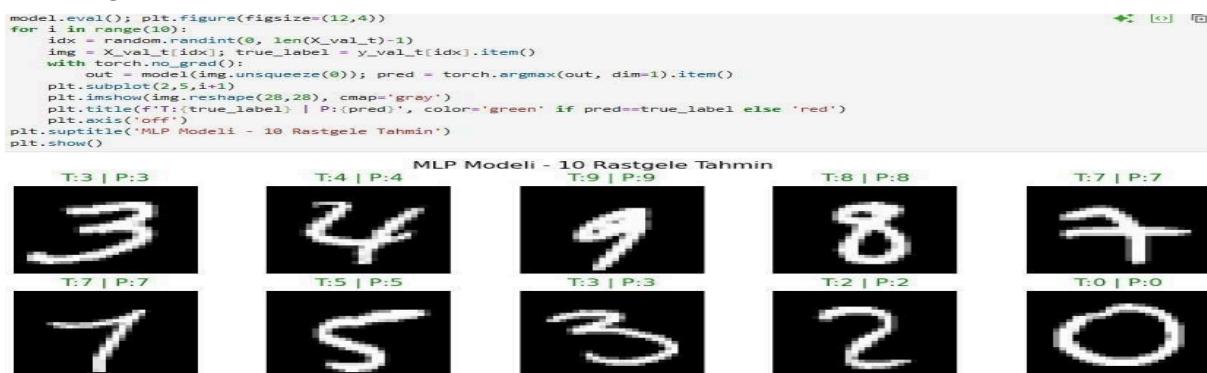
8) Kullanılan Araçlar ve Frameworkler

- Derin Öğrenme Framework'ü: PyTorch (<https://pytorch.org/>)
 - MLP ve CNN mimarilerinin tanımlanması, eğitim döngülerinin kurulması ve verimli tensör işlemlerinin yürütülmesi için kullanılmıştır.
- Yardımcı Kütüphaneler:
 - NumPy: Sayısal işlemler ve temel dizi manipülasyonları.
 - Pandas: MNIST veri setinin (CSV) yüklenmesi ve işlenmesi.
 - Matplotlib: Performans grafiklerinin (Doğruluk/Kayıp eğrileri) görselleştirilmesi.
- Geliştirme Ortamı: Tüm kodlar Jupyter Notebook formatında hazırlanmıştır.

9) Sonuçlar ve Bulgular

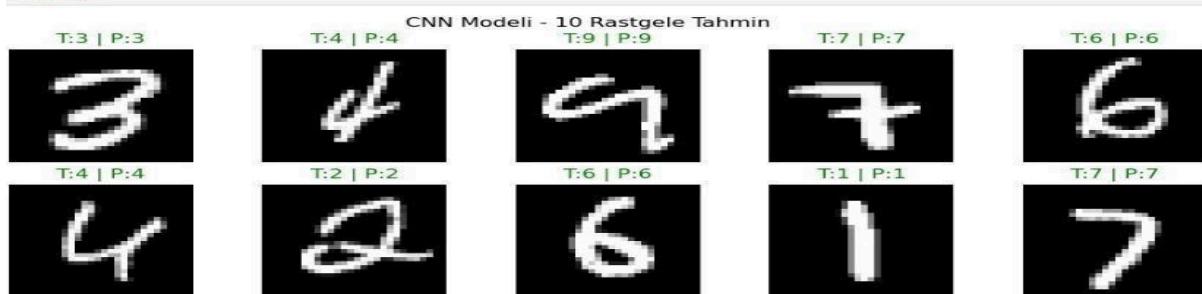
Modellerin doğrulama veri seti üzerindeki sınıflandırma başarısını görsel olarak değerlendirmek için rastgele seçilen 10 örneğe ait gerçek etiketler (T) ve model tahminleri (P) aşağıda sunulmuştur

MLP MODELİ

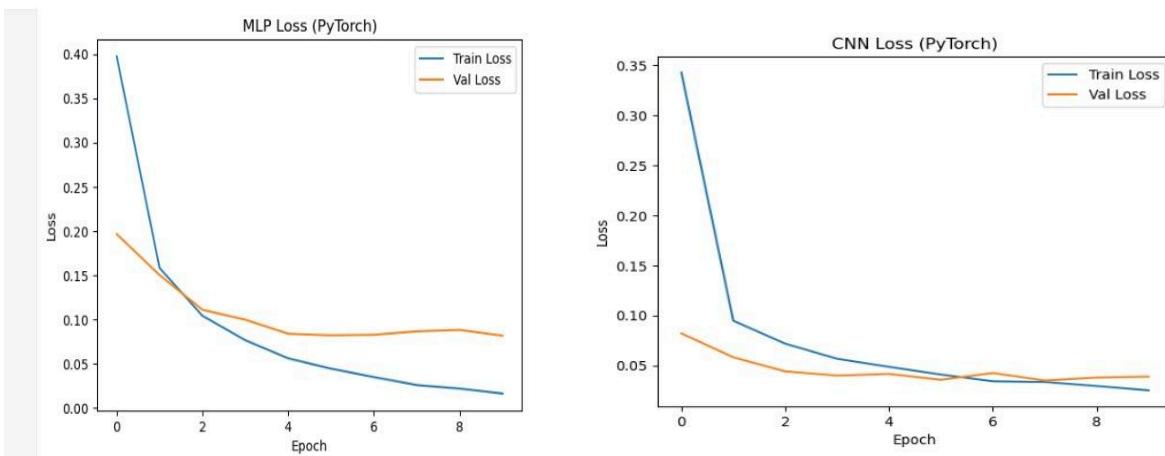


CNN MODELİ

```
model.eval(); plt.figure(figsize=(12,4))
for i in range(10):
    idx = random.randint(0, len(X_val_t)-1)
    img = X_val_t[idx]; true_label = y_val_t[idx].item()
    with torch.no_grad():
        out = model(img.unsqueeze(0)); pred = torch.argmax(out, dim=1).item()
    plt.subplot(2,5,i+1)
    plt.imshow(img.squeeze().reshape(28,28), cmap='gray')
    plt.title(f'T:{true_label} | P:{pred}', color='green' if pred==true_label else 'red')
    plt.axis('off')
plt.suptitle('CNN Modeli - 10 Rastgele Tahmin')
plt.show()
```



LOSS GRAFİKLERİ



SINIFLANDIRMA RAPORLARI

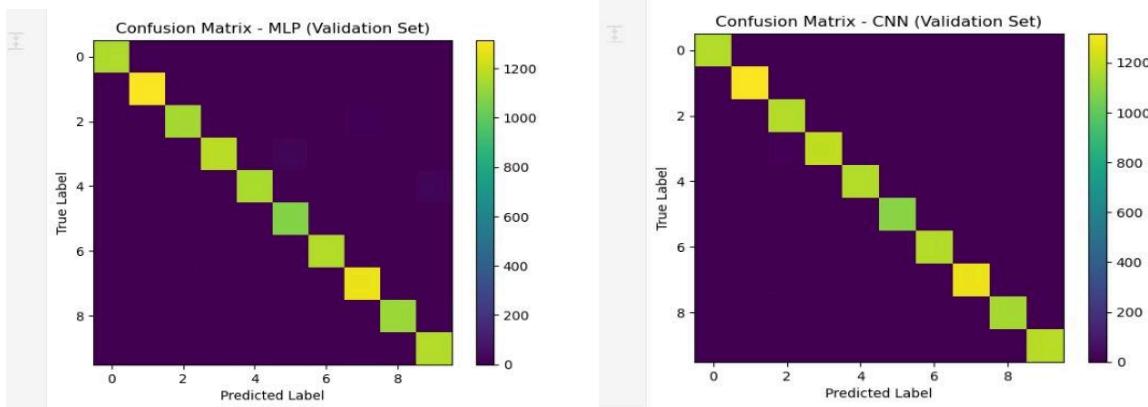
Validation Accuracy: 0.9793
Validation Precision (macro): 0.9792
Validation Recall (macro): 0.9791
Validation F1-score (macro): 0.9791

Classification Report:				
	precision	recall	f1-score	support
0	0.99	0.98	0.99	1175
1	0.99	0.99	0.99	1322
2	0.98	0.97	0.98	1174
3	0.99	0.97	0.98	1219
4	0.98	0.98	0.98	1176
5	0.97	0.98	0.97	1104
6	0.98	0.99	0.98	1177
7	0.98	0.98	0.98	1299
8	0.98	0.96	0.97	1160
9	0.97	0.98	0.97	1194
accuracy			0.98	12000
macro avg	0.98	0.98	0.98	12000
weighted avg	0.98	0.98	0.98	12000

Validation Accuracy: 0.9902
Validation Precision (macro): 0.9902
Validation Recall (macro): 0.9902
Validation F1-score (macro): 0.9902

Classification Report:				
	precision	recall	f1-score	support
0	0.99	1.00	0.99	1175
1	0.99	1.00	0.99	1322
2	0.98	0.99	0.98	1174
3	1.00	0.98	0.99	1219
4	0.99	0.99	0.99	1176
5	0.99	0.98	0.99	1104
6	0.99	1.00	0.99	1177
7	0.99	0.99	0.99	1299
8	0.99	0.99	0.99	1160
9	0.99	0.99	0.99	1194
accuracy			0.99	12000
macro avg	0.99	0.99	0.99	12000
weighted avg	0.99	0.99	0.99	12000

CONFUSION MATRIXLERİ



Mimarilerimizin Performans Karşılaştırılması

Performans Metriği	MLP Modeli (Hasan Önder Sert)	CNN Modeli (Apdullah Yuşa Önder)
Doğruluk (Accuracy)	0.9793 (%97.93)	0.9902 (%99.02)
F1-Score (Macro Avg)	0.9791	0.9902
Precision (Kesinlik)	0.9792	0.9902
Recall (Duyarlılık)	0.9791	0.9902
Final Validation Loss	0.0817	0.0388

Tartışma ve Performans Değerlendirmesi:

Elde edilen deneysel sonuçlar incelendiğinde, CNN mimarisinin (%99.02), MLP mimarisine (%97.93) kıyasla daha yüksek bir sınıflandırma başarısı gösterdiği ve hata oranını (Loss: 0.0388) MLP'ye göre (Loss: 0.0817) yarı yarıya düşürdüğü görülmüştür.

- Mimari Üstünlük: MLP modeli, 28x28 boyutundaki görüntüleri tek boyutlu bir vektöre (784 piksel) düzleştirecek (flatten) istediği için pikseller arasındaki mekansal (spatial) ilişkileri kaybetmiştir. Buna karşın CNN modeli, kullandığı evrişim (convolution) katmanları sayesinde görüntünün 2 boyutlu yapısını koruyarak kenar, eğim ve yuvarlaklık gibi yerel öznitelikleri başarıyla çıkarmıştır.

- Kararlılık: F1-Score, Precision ve Recall değerlerinin CNN modelinde birbirine çok yakın ve yüksek olması (0.9902), modelin tüm sınıflar (0-9 arası rakamlar) için dengeli ve kararlı bir öğrenme gerçekleştirdiğini kanıtlamaktadır.

10) Gelecek Çalışmalar ve Planlanan Gelişmiş Mimariler

Projede performansı artırmak ve daha karmaşık senaryoları test etmek için şu stratejiler izlenecektir:

1. Daha Derin Mimariler (ResNet/VGG): Mevcut CNN modelimiz 2 konvolüsyon bloğundan oluşmaktadır. Daha karmaşık el yazısı varyasyonlarını öğrenebilmek için katman sayısı artırılacaktır. Ancak derin ağlarda oluşabilecek "Vanishing Gradient" (Kaybolan Gradyan) problemini önlemek adına, *Skip Connection* yapısını kullanan ResNet-18 mimarisini entegre edilecektir.
2. Veri Artırma (Data Augmentation): Modelin ezberlemesini (overfitting) önlemek ve genellemeye yeteneğini artırmak için eğitim verisi üzerinde yapay çoğaltma yapılacaktır. `torchvision.transforms` kullanılarak görüntülere rastgele döndürme (rotation), kaydırma (translation) ve yakınlaştırma (zoom) işlemleri uygulanacaktır.
3. Hata Analizi (Confusion Matrix): Kod çıktılarında elde edilen Confusion Matrix (Karmaşıklık Matrisi) incelendiğinde, modellerin en çok karıştırıldığı rakamlar tespit edilip (örneğin 9 ve 4 gibi), bu sınıflara özel ağırlıklandırma veya veri temizliği yapılacaktır.

11) Kaynaklar

A. Temel Literatür ve Referans Çalışmalar

1. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998. (LeNet-5'i tanıtan ve CNN'lerin temelini atan makale).
2. G. E. Hinton, S. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, Jul. 2012. (Dropout gibi düzenlileştirme teknikleri üzerine temel çalışma).

B. Veri Seti Kaynağı

1. O. Rationale, "MNIST in CSV," Kaggle. [Çevrimiçi]. Erişim adresi: <https://www.kaggle.com/datasets/oddrationale/mnist-in-csv>. [Erişim Tarihi: 21 Kasım 2025].

C. Geliştirilen Kodlar ve Kütüphaneler

Bu kısım, geliştirilen kodların ve kullanılan temel yazılım araçlarının referanslarını içermektedir.

1. A. Y. Önder, "FET312_22040101029_DeepLeaders_1.ipynb": MNIST MLP Model Kodu," GitHub Deposu. [Çevrimiçi]. Erişim adresi: https://github.com/yusaonder/minist_deeplearning/blob/main/FET312_22040101029_DeepLeaders.ipynb
2. H. Ö. Sert, "FET312_21040301077_DeepLeaders_1.ipynb": MNIST CNN Model Kodu," GitHub Deposu. [Çevrimiçi]. Erişim adresi: https://github.com/yusaonder/minist_deeplearning/blob/main/FET312_21040301077_DeepLeaders.ipynb
3. A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Adv. Neural Inf. Process. Syst., vol. 32, pp. 8024-8035, 2019. (Projede kullanılan temel derin öğrenme kütüphanesi).