

Bileşen Açıklamaları

1. config Class

Tüm hiperparametrelerin tek bir kaynağa bağlı olmasını sağlar.

Avantajları:

- Hiperparametreler tek yerden kontrol edilir.
 - Kod içinde her fonksiyona parametre taşımaya gerek kalmaz.
 - Hem Trainer hem Model aynı kaynaktan beslenir (tutarlılık).
-

2. mycnn Class

Bu dosya sadece **model mimarisini** içerir.

- Katmanlar
- Aktivasyon fonksiyonları
- İleri geçiş (forward) fonksiyonu

Her şey bu sınıfıta düzenlenir.

3. trainer Class

Bu dosya tüm eğitim sürecinin yönetildiği yerdır.

İçerik:

- Eğitim döngüsü (train loop)
 - Değerlendirme döngüsü (validation)
 - Kayıpların hesaplanması
 - Öğrenme oranı yönetimi
 - Model kaydetme / yükleme işlemleri
-

4. utils Klasörü

data_utils.py

Sadece **data loading** (PyTorch Dataset + DataLoader) işlemleri burada yapılır.

metrics.py

Accuracy, F1-score, confusion matrix gibi metriklerin hesaplandığı yer.

misc.py

Genel yardımcı fonksiyonlar:

- dosya yolları
 - zaman hesaplamaları
 - seed belirleme
 - vb.
-

5. main.py

Projenin **merkezi yönetim dosyasıdır.**

Görevleri:

- Config yüklemeye
 - Model + Trainer oluşturma
 - Bayes optimizasyonu başlatma
 - Eğitim pipeline'ını çalıştırma
-

BAYES OPTİMİZASYONU

Parametre Optimizasyonunda Bayes Yaklaşımı (Bayesian Optimization)

I. Amaç ve Kullanım Alanı

✓ Fonksiyon değerlendirmesi pahalı olduğunda kullanılır.

Örnek:

- Büyük modeller eğitmek
- Fiziksel deneyler
- Veri toplamanın pahalı olduğu durumlar

✓ Gradyan gerektirmez.

Sadece fonksiyonun değeri $f(x)$ gereklidir.

✓ En çok hiperparametre ayarı için kullanılır.

II. Bayes Yaklaşımının Mantığı

Geleneksel yöntemler doğrudan optimumu ararken, Bayes yaklaşımı fonksiyonun **olasılıksal bir modelini** kurar.

Döngü:

1. **Gözlem:** Daha önce denenmiş parametreler alınır.
 2. **Posterior:** Bu verilere göre GP modeli güncellenir.
 3. **Acquisition Function:** Bir sonraki denenmesi gereken en iyi nokta seçilir (ör: UCB, EI, PI).
 4. **Yeni deney:** Model bu noktada eğitilir.
-

GAUSS SÜREÇLERİ (GP)

1. GP Nedir?

Bir fonksiyonun GP'den geldiğini söylemek demek:

👉 Herhangi bir sayıda noktanın çıktı değerlerinin **çok değişkenli bir Gauss dağılımı** göstermesidir.

GP'yi Tanımlayan İki Fonksiyon

1. Ortalama Fonksiyonu

$$m(x) = \mathbb{E}[f(x)]$$

2. Kovaryans (Kernel) Fonksiyonu

$$k(x, x')$$

Bu, iki noktanın fonksiyon değerlerinin **ne kadar ilişkili olduğunu** belirler.

RBF (Gaussian) Kernel

Tanım

$$k(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right)$$

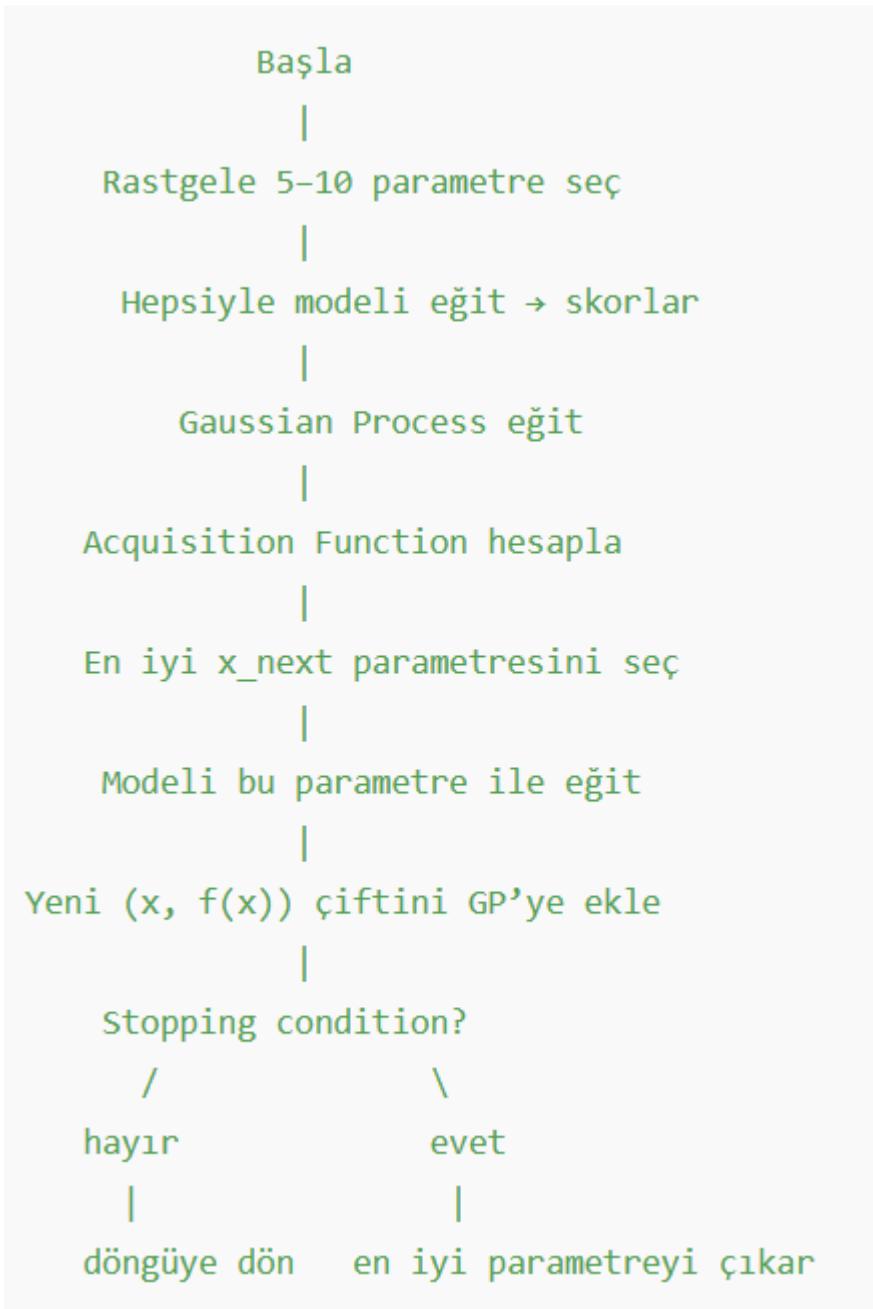
Parametreler

- σ : Fonksiyon çıktısının genel ölçüği
- l (**length scale**): Fonksiyonun ne kadar “kıvrımlı” olduğunu belirler

Küçük $l \rightarrow$ fonksiyon çok kıvrımlı

Büyük $l \rightarrow$ fonksiyon daha smooth

Bayesian Hiperparametre Seçimi Süreci



Parametre	Dataset'e Göre Değişir mi?	Açıklama
Learning rate	✓	Dataset büyüklüğüne göre ayarlanır
Batch size	✓	Dataset + GPU belleğine bağlı
Dropout	✓	Overfit durumuna göre değişir
Filters	✓	Dataset karmaşıklığı ile doğru orantılı
Kernel size	✗	Genelde sabit (3 veya 5)
Weight decay	✗	Evrensel değerler çoğu zaman yeterli
Optimizer	✓	Model mimarisi + data'ya göre

MyCNN Mimarisi

1) Model Tipi

Derin Evrişimsel Sinir Ağı

(Deep Convolutional Neural Network – CNN)

Amaç: CIFAR-10 görüntülerini sınıflandırma

2) Temel Tanımlar ve Fonksiyonlar

Yapı	Tanım / Görev
Conv2d	Görüntüden yerel desenleri öğrenen evrişimsel katman
BatchNorm2d	Aktivasyon dağılımını stabilize eder, eğitimi hızlandırır
ReLU	Doğrusal olmayanlık ekler $\rightarrow (\max(0, x))$
MaxPool2d(2)	Uzamsal boyutu yarıya indirir ($H/2, W/2$)
Dropout(p)	Aşırı öğrenmeyi azaltır, her forward'da nöronların p oranını 0 yapar
Flatten	3D tensörü ($C \times H \times W$) \rightarrow tek boyuta çevirir
Linear	Tam bağlantılı katman

Ağırlık Başlatma – `init_weights(m)`

- Conv2d & Linear katmanları **Xavier (Glorot) Initialization** ile başlatılır
 - Tüm bias'lar **0** yapılır
 - `self.apply(init_weights)` \rightarrow tüm alt modüllere özyinelemeli şekilde uygulanır
-

3) Hiperparametreler (BO Çıktıları)

Parametre	Görev	Değer
filters1	BLOK 1 konvolüsyon kanalları	83
filters2	BLOK 2 konvolüsyon kanalları	223
dropout_rate	Dropout oranı	0.222
learning_rate	Öğrenme hızı	0.00455

batch_size	Mini-batch büyülüğu	251
optimizer	Kullanılan optimizasyon algoritması	SGD
weight_decay	L2 düzenleme katsayısı	0.00043

4) Boyut Değişim Formülleri

Evrişim (Conv2d) Boyut Formülü

Kernel = 3, Padding = 1, Stride = 1 olduğundan:

$$H_{out} = \left\lfloor \frac{H_{in} - K + 2P}{S} \right\rfloor + 1 = H_{in}$$

Yani Conv → uzamsal boyut değişmez.

MaxPool2d(2)

$$H_{out} = \frac{H_{in}}{2}$$

Örn:

- $32 \times 32 \rightarrow 16 \times 16$
 - $16 \times 16 \rightarrow 8 \times 8$
-

5) BLOK 1 – Detaylı Akış (filters1 = 83)

Katman	Giriş Boyutu	Çıkış Boyutu
Giriş	$251 \times 3 \times 32 \times 32$	—
Conv2d(3→83)	$251 \times 3 \times 32 \times 32$	$251 \times 83 \times 32 \times 32$
BN → ReLU	$251 \times 83 \times 32 \times 32$	Aynı
Conv2d(83→83)	$251 \times 83 \times 32 \times 32$	$251 \times 83 \times 32 \times 32$
BN → ReLU	$251 \times 83 \times 32 \times 32$	Aynı
MaxPool2d(2)	$251 \times 83 \times 32 \times 32$	$251 \times 83 \times 16 \times 16$

6) BLOK 2 – Detaylı Akış (filters2 = 223)

Katman	Giriş Boyutu	Çıkış Boyutu
Giriş	$251 \times 83 \times 16 \times 16$	—
Conv2d(83→223)	$251 \times 83 \times 16 \times 16$	$251 \times 223 \times 16 \times 16$
BN → ReLU	$251 \times 223 \times 16 \times 16$	Aynı
Conv2d(223→223)	$251 \times 223 \times 16 \times 16$	$251 \times 223 \times 16 \times 16$
BN → ReLU	$251 \times 223 \times 16 \times 16$	Aynı
MaxPool2d(2)	$251 \times 223 \times 16 \times 16$	$251 \times 223 \times 8 \times 8$
Dropout(0.222)	$251 \times 223 \times 8 \times 8$	Aynı

7) CLASSIFIER – Tam Bağlantılı Katmanlar

Katman	Giriş	Çıkış	Açıklama
Flatten	$251 \times 223 \times 8 \times 8$	251×14272	$223 \times 8 \times 8 = 14272$
Linear(14272→256)	251×14272	251×256	1. FC
ReLU → Dropout(0.222)	251×256	251×256	Aktivasyon + Regularization
Linear(256→10)	251×256	251×10	Çıkış sınıfları (10 adet)

8) Nihai Çıktı

Her bir örnek için:

$$\hat{y} \in \mathbb{R}^{10}$$

Yani:

Her görüntü için 10 sınıf logits değeri üretilir.
Softmax sonrası bunlar sınıf olasılığına dönüşür.