

YakupLab

GPU hızlandırmalı yapay zekâ model eğitimi ve uzaktan geliştirme için Linux tabanlı iş istasyonu

Ubuntu Server	NVIDIA CUDA	PyTorch	Remote Development	AI / RL
---------------	-------------	---------	--------------------	---------

Proje Sahibi	Yakup Tekeli
Belge Türü	Teknik Proje Raporu
Durum	Tamamlandı
Tarih	Temmuz 2026

Proje Özeti

YakupLab, makine öğrenmesi, bilgisayarlı görü ve pekiştirmeli öğrenme modellerinin geliştirilmesi, eğitilmesi ve doğrulanması için kurulan kişisel bir yapay zekâ çalışma altyapısıdır. Sistem; Linux sunucu işletim sistemi, NVIDIA GPU hızlandırma, grafik masaüstü erişimi, güvenli uzaktan bağlantı ve tekrarlanabilir deney yönetimini tek bir iş istasyonunda birleştirir.

1. Amaç ve Tasarım Hedefleri

- Yerel donanım üzerinde GPU hızlandırmalı model eğitimi gerçekleştirmek.
- MacOS ve diğer istemci cihazlardan güvenli uzaktan geliştirme imkânı sağlamak.
- Uzun süreli eğitim süreçlerini masaüstü oturumundan bağımsız ve kararlı biçimde çalıştırmak.
- Python, PyTorch, CUDA, Gymnasium ve deney araçları için merkezi bir çalışma ortamı oluşturmak.
- Proje kodu, veri, model ağırlıkları, loglar ve değerlendirme çıktıları için düzenli bir dosya yapısı sağlamak.

2. Sistem Mimarisi

İstemci	Güvenli Erişim	İşletim Sistemi	Hesaplama	Deney Katmanı
Mac / dizüstü / tarayıcı →	SSH, Tailscale, web tabanlı uzak masaüstü →	Ubuntu Server + XFCE →	NVIDIA RTX 3060, CUDA, PyTorch →	Kod, veri, modeller, loglar

Mimari, komut satırı tabanlı sunucu yönetimi ile gerektiğinde kullanılan hafif XFCE masaüstünü birlikte sunar. Eğitim süreçleri doğrudan Linux üzerinde çalışırken istemci bilgisayar yalnızca geliştirme, izleme ve yönetim terminali olarak kullanılır.

3. Donanım ve Yazılım Bileşenleri

Bileşen	Teknik Açıklama
İşletim sistemi	Ubuntu Server tabanlı Linux kurulumu; kararlı servis yönetimi ve düşük sistem yükü.
Masaüstü ortamı	XFCE; uzak masaüstü oturumlarında düşük kaynak tüketimi için tercih edilmiştir.
GPU	NVIDIA GeForce RTX 3060 6 GB; CUDA destekli eğitim ve çıkarım görevleri.
AI çerçevesi	PyTorch tabanlı model geliştirme; CPU/GPU cihaz seçimi ve CUDA hızlandırması.
RL araçları	Gymnasium tabanlı özel ortamlar ve PPO/MAPPO deneyleri için çalışma ortamı.
Geliştirme	Python 3.11 sanal ortamları, Git tabanlı sürüm kontrolü ve çapraz platform kod akışı.
Uzak erişim	SSH terminal erişimi, Tailscale özel ağı ve grafik arayüz gerektiren işler için uzak masaüstü.
Gözlemlleme	Terminal tabanlı süreç takibi, sistem kaynak izleme, eğitim logları ve deney artefaktları.

4. Yapılandırma ve Uygulama Ayrıntıları

4.1 Linux ve Servis Katmanı

- Sunucu servisleri sistem açılışında otomatik çalışacak şekilde yapılandırılmıştır.
- Grafik arayüz, hesaplama altyapısından bağımsız tutulmuş ve yalnızca gerektiğinde kullanılmaktadır.
- Ağ bağlantıları ve uzaktan erişim kanalları yeniden başlatma sonrasında kalıcı olacak biçimde düzenlenmiştir.

4.2 GPU Hızlandırılmalı Eğitim Ortamı

- NVIDIA sürücüsü ve CUDA uyumlu PyTorch kurulumu ile GPU erişimi doğrulanmıştır.
- Projeler için bağımsız Python sanal ortamları kullanılarak bağımlılık çakışmaları önlenmiştir.
- Model, konfigürasyon, checkpoint ve değerlendirme çıktıları deney kimliğine göre ayrıştırılmıştır.
- MacOS geliştirme ortamı ile Linux eğitim ortamı arasındaki farklar dokümente edilmiş ve Git üzerinden senkronize edilmiştir.

4.3 Dosya ve Deney Organizasyonu

```
workspace/
  projects/<project_name>/
  datasets/
  experiments/<run_id>/
    config/
    checkpoints/
    logs/
    evaluation/
  shared_tools/
```

Bu yapı, deneylerin geriye dönük incelenmesini ve aynı konfigürasyonla yeniden çalıştırılmasını kolaylaştırır.

5. Kullanım Senaryoları

Kullanım Alanı	Uygulama
Bilgisayarlı görü	YOLO tabanlı nesne tespiti/segmentasyon, OCR modelleri ve görüntü işleme deneyleri.
Pekiştirmeli öğrenme	Özel Gymnasium ortamlarında PPO/MAPPO eğitimi, reward analizi ve politika karşılaştırması.
Model doğrulama	Checkpoint değerlendirme, çıkarım testi, metrik üretimi ve sonuçların karşılaştırılması.
Uzun süreli eğitim	İstemci bağlantısı kesilse dahi Linux üzerinde devam eden eğitim süreçleri.
Uzaktan geliştirme	Kod düzenleme, terminal erişimi, log inceleme ve dosya aktarımının özel ağ üzerinden yürütülmesi.

6. Güvenlik ve Erişim Modeli

- Sunucuya erişim, herkese açık bir masaüstü servisi yerine güvenli özel ağ ve kimlik doğrulamalı bağlantılar üzerinden sağlanır.
- Yönetim işlemleri SSH ile, grafik arayüz gerektiren işlemler ise kontrollü uzak masaüstü oturumlarıyla gerçekleştirilir.
- Proje depoları ve deney çıktıları kullanıcı bazlı Linux izinleri altında tutulur.
- İşletim sistemi ve geliştirme bağımlılıkları düzenli olarak güncellenebilir, proje ortamları ise sanal ortamlarla izole edilir.

7. Doğrulama ve Tamamlanan Çıktılar

- GPU destekli PyTorch çalışma ortamı kurulmuş ve model eğitimi için doğrulanmıştır.
- SSH, özel ağ ve grafik uzak masaüstü erişimi çalışır durumda yapılandırılmıştır.
- MacOS geliştirme - Linux eğitim iş akışı uygulanmış ve sürüm kontrolü üzerinden standardize edilmiştir.
- Pekiştirmeli öğrenme ve bilgisayarlı görü projeleri için tekrar kullanılabilir deney dizin yapısı oluşturulmuştur.
- Ses, ağ, Bluetooth, masaüstü ve gözlemlene bileşenleri günlük geliştirme kullanımına uygun hâle getirilmiştir.

8. Sonuç

YakupLab, genel amaçlı bir kişisel bilgisayardan farklı olarak yapay zekâ modeli eğitimi, deney tekrarlanabilirliği ve uzaktan çalışma gereksinimleri etrafında tasarlanmış tamamlanmış bir geliştirme altyapısıdır. Sistem, GPU hesaplama kaynaklarını merkezi hâle getirirken hafif Linux masaüstü, güvenli erişim ve düzenli deney yönetimini tek platformda birleştirir.