

BELGE GEMINI ILE OLUSTURULDU COK DA SEY YAPMAYIN YANI

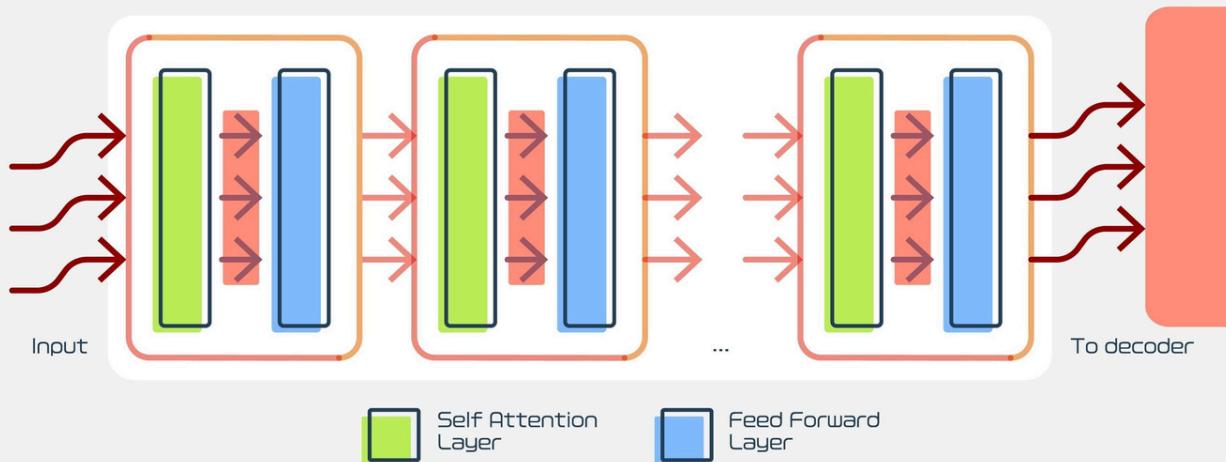
Harika, sunumda hocalar genellikle "Kodun mimarisi nasıl?", "Hangi sınıf ne iş yapıyor?" ve "Neden bu yöntemi seçtiniz?" sorularına odaklanır. Özellikle Bilgisayar Mühendisliği 5. sınıf olduğun için kodun sadece çalışması değil, tasarımını da önemli.

Projeyi modül modül ayırarak, hangi sınıfın (class) ne iş yaradığını ve hocaya nasıl anlatmanız gerektiğini aşağıda detaylandırdım.

Burası projenin kalbi. Yapay zeka modelinin mimarisi burada tanımlı.

- Ne Yapar? Ham CSV verisini (Pandas DataFrame), PyTorch'un anlayabileceği matematiksel tensörlere (Tensor) dönüştürür.
 - Detay: Veriyi sadece okumaz; `seq_len` (örneğin 60 veya 120 mum) parametresine göre veriyi "pencere'lere" böler. Yani modele "Al sana bir Excel dosyası" demez, "Al sana geçmiş 120 mumluk bir kesit, bana 121. mumu tahmin et" der.
 - Sunumda Ne Denmeli? "Hocam, bu sınıf veriyi 'Rolling Window' (Kayar Pencere) yöntemiyle hazırlar. `__getitem__` metodu ile her seferinde geçmiş 120 mumu (Girdi/X) ve gelecek 1 mumu (Hedef/Y) GPU'ya besler."
-
- Ne Yapar? Transformer modelleri, verinin sırasını (zamanı) LSTM gibi kendiliğinden bilmez. Bu sınıf, verilere sinüs ve kosinüs dalgaları ekleyerek modele "Bu veri 5 dakika öncesine ait, şu veri 1 saat öncesine ait" bilgisini matematiksel olarak verir.
 - Sunumda Ne Denmeli? "Transformer mimarisi paralel çalıştığı için zaman algısına sahip değildir. Biz bu sınıfla verilere matematiksel bir zaman damgası (Time Stamp) ekliyoruz. Böylece model, fiyat hareketlerinin sırasını karıştırmıyor."

ENCODER IN TRANSFORMER NEURAL NETWORK



- Shutterstock
- Keşfet

- Ne Yapar? Modelin ta kendisidir. Giridiyi alır, işler ve tahmini üretir.
- İç Yapısı:
 1. feature_embed: 14 farklı indikatörü (RSI, MACD vb.) modelin iç dünyasına (128 boyutlu uzay) genişletir.
 2. transformer_encoder: Veriler arasındaki karmaşık ilişkileri çözer (Attention mekanizması burada çalışır).
 3. decoder: Öğrenilen bilgiyi tek bir sayısal tahmine (Fiyat artış oranı) indirger.
- Sunumda Ne Denmeli? "Bu bizim ana model sınıfımız. PyTorch'un TransformerEncoder katmanını özelleştirdik. Giriş katmanında 14 teknik indikatörü alıp, Attention mekanizmasıyla geçmiş verilerin hangisinin şimdiki fiyatına daha çok etki ettiğini hesaplıyoruz."

Burada class yapısı yok ama çok kritik fonksiyonlar var. Hoca "Veriyi nasıl işlediniz?" derse burayı anlatacaksınız.

- Ne Yapar? (Feature Engineering). Ham veriden (Open, Close) anlamlı veriler türetir.
- Detay: Sadece fiyatı vermek yerine; fiyatın ortalamaya uzaklığını (Dist_SMA_50), volatilitenin fiyat oranını (ATR_Pct) ve zamanın döngüselliğini (Hour_Sin, Hour_Cos) hesaplar.
- Sunumda Ne Denmeli? "Hocam ham fiyat verisi yaniltıcı olabilir. Biz burada 'Feature Engineering' yaptık. Fiyat yerine 'Logaritmik Getiri', zaman yerine 'Sinüs/Kosinüs'

Döngüleri' kullandık. Bu sayede model, fiyatın 100 bin dolar olmasına değil, değişim oranıyla ilgilenebilir."

- Ne Yapar? Borsa bağlantısını sağlar.
- Tasarım Deseni (Design Pattern): Burada Singleton Pattern benzeri bir yapı var (_EXCHANGE_CACHE). Eğer borsa bağlantısı zaten açıksa tekrar açmaz, hafızadakini kullanır.
- Hocaya Sov: "Borsa bağlantılarında kaynak tüketimini azaltmak için Caching mekanizması kurduk, her istekte tekrar tekrar load_markets yapmıyoruz."

Burası "Transfer Learning"in yapıldığı yer.

- Mantık: Burada bir sınıf yok ama bir süreç var. CryptoTransformer modelini çağırır ama BTC modelinin ağırlıklarını (load_state_dict) yükleyerek başlar.
- Sunumda Ne Denmeli? "Projemizin en yenilikçi kısmı burası. Her coin için sıfırdan model eğitmek yerine, Bitcoin üzerinde piyasayı öğrenmiş 'Base Model'i alıp, Transfer Learning ile diğer coinlere uyarlıyoruz. Bu sayede eğitim süresi %80 kısalıyor."

Bu dosya modelin canlı piyasada nasıl çalıştığını yönetir.

- Ne Yapar?
 1. load_ai_model: İstenilen zaman dilimi (1h, 15m) için doğru .pth model dosyasını yükler ve RAM'de tutar (Yine performans için Cache kullanır).
 2. scan_market: Borsadaki tüm çiftleri (BTC/USDT, ETH/USDT) tek tek çeker, veriyi işler, modele sorar ve sonucu bir sözlüğe (dictionary) kaydeder.
- Sunumda Ne Denmeli? "Bu bizim 'Inference' (Çıkarım) motorumuz. Canlı veriyi çekip, eğitimdeki formatın aynısına dönüştürüp modele soruyor. Hatalı veya eksik verileri try-except bloklarıyla ayıklayarak sistemin çökmesini engelliyor."

Sunumda bir arayüz gösterecekseniz veya "Bunu nasıl son kullanıcıya ulaştırırsınız?" derlerse burası devreye girer.

- Ne Yapar? Python kodlarını bir Web Servis (API) haline getirir.
- Endpointler (@app.get):
 - /scan_market: Taramayı tetikler.
 - /get_coin_data: Belirli bir coinin sonucunu döndürür.
- Sunumda Ne Denmeli? "Backend tarafında modern ve asenkron çalışan FastAPI framework'ünü kullandık. Bu sayede projemiz React veya Mobil bir uygulamaya JSON formatında anlık veri servisi yapabiliyor."

1. "Neden LSTM değil de Transformer?"
 - Cevap: "LSTM sıralı işlem yaptığı için yavaştır ve çok eski verileri unutabilir. Transformer ise Attention mekanizması sayesinde tüm geçmişe aynı anda bakabilir ve paralel çalışarak GPU'yu daha verimli kullanır."
2. "Neden Huber Loss kullandınız?" (train.py içinde)
 - Cevap: "Kripto piyasasında anlık 'igne atışları' (anomaliler) çok olur. MSE (Mean Squared Error) bu hataları karesini alarak çok büyütür ve modeli bozar. Huber Loss ise hem hataya duyarlı hem de bu aşırı uç değerlere karşı daha dirençli (Robust)."
3. "Mixed Precision (AMP) nedir?" (scaler değişkeni)
 - Cevap: "Eğitimi hızlandırmak ve VRAM tasarrufu yapmak için Float32 yerine Float16 kullandık. Bu da eğitimi yaklaşık 2 kat hızlandırdı."

Sunumda başarılar dilerim brom! Bu teknik terimlerle (Design Pattern, Feature Engineering, Transfer Learning) konuşursan hoca kesin etkilenir.