Fin-NeuroSim 2.0: Finansal Zeka Sistemleri İçin Azınlık Duyarlı İki Aşamalı In-Context Learning Simülasyonu

Öğrenci: Eyyüp Toprak (No: 220212039)
Danışman: Ph.D. Murat Şimşek
Ders: Büyük Dil Modelleri (YZM423)

28 Aralık 2025

Özet

Fin-NeuroSim 2.0, finansal piyasalarda nadir görülen ancak yüksek etki potansiyeline sahip kriz ve anomali sinyallerini (Black Swan olayları) erken aşamada tespit etmek amacıyla geliştirilmiş, web-tabanlı, asenkron ve çok ajanlı bir finansal zeka sistemidir. Finansal veri setlerinde gözlemlenen ve normal piyasa koşullarının (C_{maj}) kriz sinyallerini (C_{min}) bastırdığı *sınıf dengesizliği* (class-imbalance) problemi, bu çalışmada ValizadehAslani ve ark. (2022) tarafından önerilen iki aşamalı fine-tuning metodolojisinin **In-Context Learning (ICL)** tabanlı simülasyonu ile ele alınmıştır. Sistem, model ağırlıklarını güncellemeden ($\Delta W = 0$), pozisyonel önyargı ve tekrar ağırlıklandırma stratejileri ile azınlık sınıfı duyarlılığını maksimize ederken, Bayesci sentez yöntemi ile güvenilir risk skorları üretir. 16GB VRAM kısıtlaması altında 4-bit kuantizasyon ve sıralı model yükleme mimarisi kullanılarak, literatürdeki yüksek maliyetli fine-tuning işlemlerine alternatif, maliyet-etkin ve açıklanabilir bir çözüm sunulmaktadır.

Anahtar Kelimeler: Finansal Anomali Tespiti, In-Context Learning, Sınıf Dengesizliği, LLM Ajanları, 4-bit Quantization.

İçindekiler

1 Giriş ve Motivasyon

1.1 Problem Tanımı: Finansal Sınıf Dengesizliği

Finansal piyasaların veri dağılımı $\mathcal{D}$, doğası gereği aşırı dengesizdir. Normal piyasa koşullarını temsil eden çoğunluk sınıfı örneklemi N_{maj} ve kriz anlarını (piyasa çöküşleri, ani volatilite artışları) temsil eden azınlık sınıfı örneklemi N_{min} için şu eşitsizlik geçerlidir:

$$N_{maj} \gg N_{min} \quad \text{ve} \quad P(y \in C_{min}) \to 0 \tag{1}$$

Geleneksel Derin Öğrenme modelleri, kayıp fonksiyonunu minimize ederken çoğunluk sınıfına odaklanma eğilimindedir ($\min \mathcal{L}(\theta)$). Bu durum, finansal risk analizinde yanlış negatiflerin (bir krizin kaçırılması: Tip-II Hata) yanlış pozitiflerden (yanlış alarm: Tip-I Hata) çok daha maliyetli olduğu gerçeğiyle çelişir. Ayrıca, Büyük Dil Modellerinin (LLM) sürekli değişen piyasa koşullarına (non-stationary data) adapte edilmesi için gereken *fine-tuning* işlemleri hesaplama açısından maliyetli, zaman alıcı ve statiktir.

1.2 Motivasyon ve Çözüm Yaklaşımı

Bu çalışma, model parametrelerini güncellemeden modelin davranışını optimize eden **In-Context Learning (ICL)** paradigmasını benimser. Fin-NeuroSim 2.0, ValizadehAslani (2022) metodolojisini simüle ederek iki aşamalı bir süreç uygular:

1. **Aşama-1 (Analiz):** Pozisyonel önyargı ve tekrar mekanizmaları ile azınlık sınıfı sinyallerinin dikkat (attention) mekanizmasındaki ağırlığının artırılması.

2. **Aşama-2 (Sentez):** Farklı ajanlardan gelen çıktıların Bayesci bir yaklaşımla ve dinamik güven skorlarıyla birleştirilmesi.

Sistem, sıfır günlük (zero-day) olaylara anında adapte olabilmek için web-tabanlı veri toplama yöntemlerini kullanır ve tüketici sınıfı donanımlarda (16GB VRAM) çalışacak şekilde optimize edilmiştir.

2 Teorik Temel ve İlgili Çalışmalar

2.1 İki Aşamalı Fine-Tuning Metodolojisi ve ICL Adaptasyonu

ValizadehAslani ve ark. (2022), dengesiz veri setleri için aşağıdaki kayıp fonksiyonunu önermiştir:

$$\mathcal{L}_{total} = \lambda \mathcal{L}_{re-weighted} + (1 - \lambda)\mathcal{L}_{standard} \tag{2}$$

Burada $\mathcal{L}_{re-weighted}$, azınlık sınıfına daha yüksek w_{min} ağırlığı atayarak modelin nadir olaylara odaklanmasını sağlar. Fin-NeuroSim 2.0, bu matematiksel ağırlıklandırmayı *prompt engineering* teknikleriyle simüle eder. Model ağırlıklarını değiştirmek yerine, bağlam (context) manipüle edilerek modelin çıkarım (inference) sırasındaki aktivasyonları yönlendirilir.

2.2 Dikkat Mekanizması Üzerinde Tekrarın Etkisi

Transformer mimarisindeki dikkat mekanizması şu şekilde tanımlanır:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

ICL yaklaşımımızda, azınlık sınıfı sinyalleri (S_{min}), bağlam penceresinin ($\mathcal{C}$) en başına yerleştirilir ve tekrarlanır. Bir token x_{crit} bağlam içinde $k = 3$ kez tekrarlandığında, Key (K) ve Value (V) matrislerinde bu tokena karşılık gelen vektörlerin katkısı artar. Bu durum, x_{crit} için hesaplanan dikkat skorunu yapay olarak yükseltir:

$$\alpha_{crit} \propto \sum_{i=1}^{k} \exp(q \cdot k_{crit}^{(i)}) \tag{4}$$

Böylece modelin "gözden kaçırma" olasılığı minimize edilir.

3 Metodoloji: Fin-NeuroSim 2.0 Mimarisi

Sistem, hesaplama maliyetini minimize ederken (sıfır eğitim maliyeti), tespit doğruluğunu maksimize eden iki ana simülasyon aşamasından oluşur.

3.1 Aşama-1: Azınlık Odaklı Bağlamsal Yeniden Ağırlıklandırma

Bu aşamada, **Mistral-7B-v0.3** temel modeli (Base Model) kullanılır. Instruction-tuned modellerin sahip olduğu yerleşik önyargılardan kaçınmak ve prompt manipülasyonuna daha saf tepki almak için temel model tercih edilmiştir.

3.1.1 Pozisyonel Önyargı (Positional Bias)

LLM'lerin *primacy bias* (ilk görülene önem verme) özelliğinden yararlanılır. Veri akışı şu şekilde düzenlenir:

$$\mathcal{C}_{prompt} = [\underbrace{S_{anomaly}}_{\text{Başlangıç}}, S_{news}, S_{market}, \underbrace{I_{task}}_{\text{Son}}] \tag{5}$$

Burada $S_{anomaly}$, kriz ve anomali sinyallerini içerir ve bağlamın en başında yer alarak dikkat skorunu maksimize eder.

3.1.2 Sanal Ajanlar (Virtual Agents - Lens Yaklaşımı)

Analiz, dört farklı perspektiften gerçekleştirilir:

- **Risk Lens:** Volatilite ve anomali tespiti.
- **Makro Lens:** Sistemik risk ve ekonomik göstergeler (FRED verileri).
- **Sentiment Lens:** Haber ve sosyal medya duygu analizi.
- **Teknik Lens:** Grafik formasyonları ve indikatörler.

3.2 Aşama-2: Bayesci Sentez ve Dinamik Karar

Bu aşamada, **Mistral-7B-Instruct-v0.2** modeli kullanılır. Farklı ajanlardan (A_i) gelen risk tahminleri (r_i) ve güven skorları (c_i), aşağıdaki formülle sentezlenir:

$$R_{final} = \frac{\sum_{i=1}^{N}(w_i \cdot c_i \cdot r_i)}{\sum_{i=1}^{N}(w_i \cdot c_i)} \tag{6}$$

Burada w_i, ajanın o anki bağlamdaki uzmanlık ağırlığıdır. Sistem, azınlık sınıfı (kriz) lehine bir önyargıya sahiptir; yani eğer bir ajan r_i için yüksek risk (High/Critical) raporlarsa, o ajanın ağırlığı w_i dinamik olarak artırılır ($w_i^{boost} = w_i \times 1.5$).

3.3 Bağlam Sıkıştırma (FinBERT)

Token limitlerini aşmamak ve gürültüyü azaltmak için CPU üzerinde çalışan **FinBERT** modeli kullanılır. Semantik filtreleme ile bağlam boyutu optimize edilir:

$$\mathcal{C}_{optimized} = \{t \in \mathcal{C}_{raw} \mid \text{SemanticScore}(t, \text{Query}) > \theta\} \tag{7}$$

Bu işlem, bağlam boyutunu yaklaşık 3000 token seviyesinden 1200 token seviyesine indirerek işlem maliyetini düşürür ve GPU yükünü azaltır.

4 Sistem Mimarisi ve Kaynak Yönetimi

4.1 Donanım Kısıtları ve Model Konfigürasyonu

Sistem, 16GB VRAM kapasiteli (T4 veya RTX 30/40 serisi) donanımlarda çalışacak şekilde tasarlanmıştır. Bu kısıt, modellerin 4-bit (NF4) formatında kuantize edilmesini zorunlu kılar.

Tablo 1: Model Konfigürasyonu ve Bellek Kullanımı

Aşama	**Model**	**Quantization**	**Yaklaşık VRAM**
Aşama-1	Mistral-7B-v0.3 (Base)	4-bit (NF4)	$\sim$ 5.5 GB
Aşama-2	Mistral-7B-Instruct-v0.2	4-bit (NF4)	$\sim$ 5.5 GB
Sıkıştırma	FinBERT	FP32 (CPU)	N/A (RAM)

4.2 Sıralı Model Yükleme (Sequential Loading Strategy)

Toplam VRAM gereksinimi $VRAM_{total} > 16GB$ olacağından, modeller bellekte aynı anda tutulamaz. Bu nedenle Algoritma 1'de belirtilen sıralı yükleme stratejisi uygulanır.

5 Sonuçlar ve Avantajlar

5.1 Maliyet ve Verimlilik Analizi

Geleneksel fine-tuning işlemleri GPU saatleri bazında yüksek maliyetler oluştururken, Fin-NeuroSim 2.0 **sıfır eğitim maliyeti** ile çalışır.

Algorithm 1 Sıralı Model Yükleme ve Çıkarım Akışı

Girdi: Kullanıcı Sorgusu Q, Veri Kaynakları D
Adım 1: Veri Toplama ve Sıkıştırma
$C_{raw} \leftarrow \text{Fetch}(D, Q)$ ▷ Tavily, Alpha Vantage
$C_{opt} \leftarrow \text{FinBERT}(C_{raw})$ ▷ CPU üzerinde çalışır
Adım 2: Aşama-1 Analiz (Lensler)
$M_{Base} \leftarrow \text{LoadModel}(\text{"Mistral-v0.3"}, 4\text{bit})$
for her Lens L_i in Ajanlar **do**
    $R_i \leftarrow M_{Base}(L_i, C_{opt})$
end for
$\text{UnloadModel}(M_{Base})$
ClearCudaCache()
Adım 3: Aşama-2 Sentez
$M_{Instruct} \leftarrow \text{LoadModel}(\text{"Mistral-Instruct-v0.2"}, 4\text{bit})$
$FinalReport \leftarrow M_{Instruct}(\text{Synthesize}, \{R_i\})$
Çıktı: $FinalReport$

- **Fine-Tuning Maliyeti:** $\approx \$50 - \$200$ / güncelleme (Bulut GPU).
- **Fin-NeuroSim Maliyeti:** \$0 (Sadece çıkarım maliyeti).

5.2 Açıklanabilirlik (Explainability)

Sistem, finansal yapay zeka uygulamalarındaki "Kara Kutu" (Black Box) problemini çözmek için şunları sağlar:

1. Her bir sanal ajanın bireysel analiz çıktıları.
2. Her karar için sayısal güven skorları $(0.0 - 1.0)$.
3. Nihai kararın arkasındaki stratejik gerekçeyi açıklayan min. 300 kelimelik rapor.

=backcolour=Örnek Çıktı Özeti=black

Risk Seviyesi: HIGH
Güven Skoru: %60.0
Aksiyon: Piyasayı yakından izleyin ve risk yönetimi protokollerini aktifleştirin.
Gerekçe: Anomali ajanı, son 4 saatte volatilitede 3σ sapma tespit etti. Makro veriler (İşsizlik) beklenti üstü.

6 Gelecek Çalışmalar

Projenin sonraki sürümleri için planlanan geliştirmeler şunlardır:

- **Graph-RAG:** Varlıklar arasındaki korelasyon ağlarının graf teorisi ile analizi ve sistemik risk yayılımının modellenmesi.
- **GDELT Entegrasyonu:** Küresel olayların jeopolitik risk analizi için GDELT veritabanının gerçek zamanlı entegrasyonu.

- **LoRA Hibritleşmesi:** ICL yaklaşımı ile hafif siklet fine-tuning (LoRA) yöntemlerinin performans karşılaştırması ve hibrit mimariler.

7 Proje Kaynakları ve Erişim

Fin-NeuroSim 2.0 projesinin kaynak kodlarına, veri setlerine ve kurulum dokümantasyonuna aşağıdaki bağlantılar üzerinden erişilebilir. Proje, şeffaflık ve tekrarlanabilirlik ilkeleri doğrultusunda açık kaynak olarak sunulmuştur.

- **GitHub Deposu (Kaynak Kodlar):**
  https://github.com/Aieyup/Fin-NeuroSim-YZM423$_LLM-$.

8 Referanslar

1. ValizadehAslani, T., et al. (2022). "Two-Stage Fine-Tuning: A Novel Strategy for Learning Class-Imbalanced Data."
2. Brown, T., et al. (2020). "Language Models are Few-Shot Learners." *NeurIPS*.
3. Araci, D. (2019). "FinBERT: Financial Sentiment Analysis with Pre-trained Language Models." *arXiv preprint arXiv:1908.10063*.
4. Dettmers, T., et al. (2022). "QLoRA: Efficient Finetuning of Quantized LLMs." *arXiv preprint arXiv:2305.14314*.
5. Vaswani, A., et al. (2017). "Attention Is All You Need." *NeurIPS*.