

Kodunuzu ve isteklerinizi inceledikten sonra, kodunuzun mevcut durumunu FTMO uyumlu bir ticaret botu geliştirmek için belirlediğiniz gereksinimlerle karşılaştırdım. Kodunuz, Soft Actor-Critic (SAC) tabanlı bir reinforcement learning (RL) yaklaşımıyla sağlam bir temel sunuyor ve birçok teknik göstergeyi (RSI, EMA, MACD, ATR vb.) entegre ediyor. Ancak, isteklerinizde belirtilen bazı önemli özellikler eksik veya tam olarak uygulanmamış durumda. Aşağıda, kodunuzda eksik olanları ve iyileştirme önerilerimi detaylı bir şekilde sunuyorum.

Kodunuzda Eksik Olanlar

1. FTMO Uyumluluğu ve Risk Yönetimi

- **Eksiklik:**

- İşlem başı risk %2 (ISLEM_BASINA_MAKS_RISK = 0.02) olarak tanımlanmış, ancak FTMO kuralları işlem başı riskin %1'i geçmemesini gerektiriyor.
- Günlük %5 ve toplam %10 drawdown limitleri kodda mevcut (MAX_DAILY_LOSS ve MAX_TOTAL_LOSS), ancak backtest modunda bu limitler sırasıyla %7 ve %15 olarak ayarlanmış, bu da FTMO kurallarına uymuyor.

- **Öneri:**

- ISLEM_BASINA_MAKS_RISK değerini 0.01'e düşürün.
- MAX_DAILY_LOSS ve MAX_TOTAL_LOSS değerlerini her modda sırasıyla 0.05 ve 0.10 olarak sabitleyin.

2. Multi-Timeframe Analizi

- **Eksiklik:**

- Kod, farklı zaman dilimlerinden (M15, H1, H4) özellikler (örneğin rsi_H1, adx_H4) alıyor, ancak bu veriler tam anlamıyla bir multi-timeframe stratejisi olarak entegre edilmemiş. Yani, üst zaman dilimlerinden gelen trend doğrulaması ile alt zaman dilimlerinden giriş sinyali senkronizasyonu yok.

- **Öneri:**

- _sinyal_gucu_hesapla fonksiyonuna bir multi-timeframe filtersi ekleyin. Örneğin, H1'de RSI < 30 ve M15'te MACD kesişimi gibi koşullarla sinyal gücünü artırın.

3. Direkt Haber Filtresi

- **Eksiklik:**

- Ekonomik takvim verisi yükleniyor ve VADER ile sentiment analizi yapılıyor, ancak yüksek etkili haberler öncesi ve sonrası işlem durdurma mekanizması tam olarak uygulanmamış. _sinyal_gucu_hesapla fonksiyonunda duygusal skoru dikkate alınıyor, ama bu yeterli değil.

- **Öneri:**

- step fonksiyonuna bir haber kontrol katmanı ekleyin. Örneğin,

takvimdeki "yüksek etkili" olaylardan 30 dakika önce ve sonra işlem açmayı engelleyin.

4. Trailing Stop

- **Eksiklik:**

- Kodda sabit SL ve TP var (_islem_simule_et içinde ATR bazlı), ancak trailing stop özelliği yok. _yuzen_pnl_hesapla içinde SL'nin kârda dinamik olarak güncellenmesi için bir mantık mevcut, ama bu sınırlı ve tam bir trailing stop değil.

- **Öneri:**

- _yuzen_pnl_hesapla fonksiyonuna tam bir trailing stop mekanizması ekleyin. Örneğin, pozisyon belirli Motorik bir mesafe (örneğin ATR * 0.5) kadar kâr alındığında SL'yi giriş fiyatına çekin (breakeven).

5. Kapsamlı Raporlama

- **Eksiklik:**

- Günlük loglama yapılıyor (ticaret_bot.log), ancak günlük, haftalık, aylık PnL, winrate, drawdown, başarısız işlem nedenleri gibi detaylı raporlama eksik. Ayrıca, motivasyon mesajları veya kullanıcı dostu özetler yok.

- **Öneri:**

- _bilgi_al fonksiyonunu genişletin ve günlük/haftalık/aylık raporlar oluşturmak için bir Raporlama sınıfı ekleyin. CSV veya Telegram üzerinden detaylı özetler gönderin.

6. Manuel Müdahale

- **Eksiklik:**

- Telegram bildirimleri var, ancak kullanıcıların botu manuel olarak durdurma, risk seviyesini ayarlama gibi komutlarla kontrol etme özelliği yok.

- **Öneri:**

- TelegramBot sınıfına komut işleme ekleyin (örneğin /pause, /resume, /set_risk 0.01).

7. Gelişmiş Risk Yönetimi

- **Eksiklik:**

- Kademeli giriş/çıkış (scaling in/out), kâr kilitleme (lock profit) veya ardışık kazanç/kayıp sonrası dinamik risk ayarı gibi özellikler eksik.

- **Öneri:**

- _islem_simule_et ve _pozisyonu_kapat fonksiyonlarına kademeli giriş/çıkış mantığı ekleyin. Örneğin, kâr +20 pip'e ulaşırsa pozisyonun %50'sini kapatın.

8. Kendi Kendine Optimizasyon

- **Eksiklik:**

- RL modeli öğreniyor, ancak hiperparametrelerin otomatik optimizasyonu (self-tuning) veya başarısız alanların yanında tespiti için bir mekanizma yok.

- **Öneri:**

- Bayesian optimizasyon veya grid search ile periyodik hiperparametre tuning ekleyin.

9. Kullanıcı Özelleştirme

- **Eksiklik:**

- Kullanıcıların hedef kâr belirleme, risk seviyesini ayarlama veya strateji preset'leri seçme gibi özelleştirme seçenekleri yok.

- **Öneri:**

- argparse ile ek kullanıcı argümanları (örneğin --target_profit, --risk_level) ekleyin ve bunları TicaretOrtami sınıfında uygulayın.

10. Diğer Eksik Özellikler

- **Piyasa Rejimine Özel Stratejiler:** Rejim tespiti var (piyasa_rejimi), ancak her rejim için farklı stratejiler tanımlı değil.
- **Pozisyon Katlama:** Kazançlı serilerde lot artışı veya kademeli çıkış yok.
- **Zaman Bazlı Filtreler:** Prime zaman filtresi var, ancak Pazartesi/Cuma gibi özel gün filtreleri eksik.
- **Gelişmiş RL Algoritmaları:** Sadece SAC kullanılıyor; PPO, DDPG gibi diğer algoritmalar denenmemiş.

İyileştirme Önerilerim

Aşağıda, kodunuzu MacBook M1 üzerinde optimize etmek ve gereksinimlerinizi karşılamak için önerilerimi sıralıyorum:

1. FTMO Uyumluluğunu Sağlama

python

```
ISLEM_BASINA_MAKS_RISK = 0.01 # %1 risk
self.MAX_DAILY_LOSS = 0.05 # Günlük %5
self.MAX_TOTAL_LOSS = 0.10 # Toplam %10
```

- Tüm modlarda bu değerleri sabit tutun ve step içinde sıkı kontrol ekleyin.

2. Multi-Timeframe Stratejisi

python

```
def _sinyal_gucu_hesapla(self, sembol: str) -> float:
    oz = self.ozellikler_dict[sembol].iloc[self.mevcut_adim]
    skor = 0.0
    if oz['rsi_H1'] < 30 and oz['macd'] > oz['macd_signal']: # H1 ve M15 uyumu
        skor += 0.5
    # Mevcut diğer koşulları koruyun
    return min(skor, 1.0)
```

3. Direkt Haber Filtresi

python

```
def _haber_kontrol(self):
    if not self.takvim_df.empty:
        mevcut_zaman = self.mevcut_tarih
        haberler = self.takvim_df[(self.takvim_df['date'] >= mevcut_zaman -
pd.Timedelta(minutes=30)) &
                           (self.takvim_df['date'] <= mevcut_zaman +
```

```

pd.Timedelta(minutes=30)) &
        (self.takvim_df['önem'] == 'yüksek')]
    return len(haberler) == 0
return True

def step(self, action: np.ndarray):
    if not self._haber_kontrol():
        self.islem_yok_sayisi += 1
    return self._gozlem_al(), 0.0, self.bitti, False, self._bilgi_al(0.0)
# Mevcut step mantığını devam ettirin

```

4. Trailing Stop

python

```

def _yuzen_pnl_hesapla(self) -> float:
    toplam_pnl = 0.0
    for islem_id, poz in list(self.pozisyonlar.items()):
        mevcut_fiyat = self.df_full_dict[poz['sembol']]
        ['close'].iloc[self.mevcut_adim]
        atr = self.ozellikler_dict[poz['sembol']]['atr'].iloc[self.mevcut_adim]
        if poz['tip'] == 'buy':
            poz[' pnl'] = (mevcut_fiyat - poz['giris_fiyati']) * poz['lot_buyuklugu'] *
10000
            if poz[' pnl'] > atr * 2: # Kâr 2x ATR ise trailing stop
                poz['sl'] = max(poz['sl'], mevcut_fiyat - atr * 0.5)
            # Sell için benzer mantık
            toplam_pnl += poz[' pnl']
    return toplam_pnl

```

5. Kapsamlı Raporlama

python

class Raporlama:

```

def gunluk_rapor(self, env):
    rapor = (f"Günlük PnL: ${env.gunluk_kapali_pnl:.2f}\n"
             f"Winrate: {len([i for i in env.islemler if i[' pnl'] > 0]) / "
len(env.islemler):.2%}\n"
             f"Max DD: {env.gunluk_kayip:.2f}\n"
             f"Motivasyon: Harika iş Chief!")
    env.telegram.senkron_mesaj_gonder(rapor)

```

```

pd.DataFrame(env.islemler).to_csv(f"gunluk_islemler_{env.mevcut_tarih.date()}"
.csv")

```

6. Performans Optimizasyonu (MacBook M1)

- M1 Kullanımı:** args.cihaz = 'mps' olarak ayarlayın (Metal Performance Shaders, M1 GPU'sunu kullanır). Stable-Baselines3'ün MPS desteği sınırlı olabilir, bu yüzden kontrol edin.
- Paralelleştirme:** n_envs değerini 8-12 arasında tutun; M1'in 8 çekirdeğini verimli kullanır.
- Veri İşleme:** Polars ve NumPy zaten hızlı, ancak ozellik_muhandisligi içindeki döngülerı vektörize edin.

7. Ek Öneriler

- **Gelişmiş RL:** PPO veya DDPG'yi test etmek için SAC yerine bir ensemble yaklaşımı deneyin.
- **Backtesting:** main fonksiyonuna kapsamlı bir backtest döngüsü ekleyin (farklı yıllar ve piyasa koşulları için).
- **Kullanıcı Dostu Arayüz:** Basit bir dashboard için streamlit veya Telegram'a emoji destekli raporlar ekleyin (örneğin • / •).

Sonuç

Kodunuz, teknik analiz ve RL temellerinde güçlü bir başlangıç yapıyor, ancak FTMO uyumluluğu, gelişmiş risk yönetimi, raporlama ve kullanıcı etkileşimi gibi alanlarda eksiklikler var. Yukarıdaki önerilerle, botunuzu hem daha sağlam hem de kullanıcı dostu hale getirebilirsiniz. MacBook M1'in performansını tam kullanmak için MPS desteği ve optimizasyonlara odaklanın. Eksik özelliklerinizi adım adım uygulayarak FTMO başarınızı artırabilirsiniz!