

Attention LSTM 기반 커피 가격 예측 모델의 이론적 배경

1. 시계열 예측의 기본 원리

시계열 예측은 시간에 따라 수집된 데이터 포인트를 분석하여 미래 값을 예측하는 과정이다. 전통적인 시계열 분석 방법(ARIMA, VAR 등)은 선형성 가정에 기반하여 작동하지만, 커피 가격과 같은 실제 시장 데이터는 복잡한 비선형 패턴을 보이는 경우가 많다. 이러한 한계를 극복하기 위해 딥러닝 기반 접근법이 도입되었다.

2. LSTM(Long Short-Term Memory) 네트워크

2.1 RNN의 한계와 LSTM의 등장

일반적인 RNN(Recurrent Neural Network)은 시퀀스 데이터 처리에 사용되지만, 장기 의존성 문제(긴 시퀀스에서 초기 정보가 손실되는 현상)와 기울기 소실/폭발 문제를 갖고 있다. LSTM은 이러한 문제를 해결하기 위해 특별히 설계된 RNN의 변형이다.

2.2 LSTM의 구조와 작동 원리

LSTM의 핵심은 세 가지 게이트를 통해 정보 흐름을 제어하는 셀 상태(cell state)다:

1. **망각 게이트(Forget Gate)**: 이전 정보 중 어떤 것을 버릴지 결정함

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

2. **입력 게이트(Input Gate)**: 새로운 정보 중 어떤 것을 저장할지 결정함

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \end{aligned}$$

3. **출력 게이트(Output Gate)**: 셀 상태의 어떤 부분을 출력할지 결정함

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned}$$

4. **셀 상태 업데이트**:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

이 구조는 장기 의존성을 보존하면서도 관련 없는 정보를 필터링할 수 있게 해준다.

2.3 다층 LSTM의 효과

이 프로젝트에서는 다층 LSTM(num_layers=2)을 사용한다. 각 층은 이전 층의 추상화된 표현을 입력으로 받아 더 복잡한 패턴을 학습할 수 있게 한다. 첫 번째 층이 로컬 패턴을 포착한다면, 두 번째 층은 이러한 패턴 간의 관계를 학습할 수 있다.

3. 어텐션 메커니즘

3.1 어텐션의 기본 개념

어텐션 메커니즘은 "모든 입력이 동등하게 중요하지 않다"는 아이디어에서 출발한다. 입력 시퀀스의 각 요소에 대해 상대적 중요도를 계산하고, 이를 가중치로 사용하여 컨텍스트 벡터를 생성한다.

3.2 어텐션 계산 과정

1. **점수 계산**: 각 시간 단계 t 에 대한 점수 계산

```
score(t) = alignment_model(hidden_state, encoder_output(t))
```

2. **가중치 변환**: 점수를 확률 분포로 변환

```
weights = softmax(scores)
```

3. **컨텍스트 벡터 생성**: 가중치를 사용한 가중합

```
context = sum(weights * encoder_outputs)
```

3.3 Entmax와 희소 어텐션

Softmax의 한계

전통적인 softmax는 모든 입력에 양의 가중치를 할당한다. 이는 시계열 데이터에서 모든 시점이 예측에 영향을 미치게 되어, 가격 변동성이 큰 기간에 대한 예측이 평탄화되는 문제를 야기할 수 있다.

Entmax15의 이론적 배경

Entmax15는 스파스 어텐션(sparse attention)을 위한 활성화 함수로, α -entmax 패밀리의 일종이다. $\alpha=1.5$ 일 때의 특별한 경우가 Entmax15다.

Entmax는 다음 최적화 문제의 해다:

$$\text{entmax}_{\alpha}(x) = \operatorname{argmax}_p \langle p, x \rangle - H_{\alpha}(p)$$

여기서 H_α 는 Tsallis α -entropy다.

Entmax의 주요 특성

1. **희소성(Sparsity)**: 중요하지 않은 입력에 정확히 0의 가중치를 할당한다.
2. **연속성(Smoothness)**: 미분 가능하여 역전파를 통한 학습이 가능하다.
3. **매개변수화(Parameterization)**: α 매개변수를 통해 희소성의 정도를 조절할 수 있다.

커피 가격 예측에서의 Entmax 효과

커피 가격은 기후 이벤트, 지정학적 상황, 시장 심리 등의 급격한 변화에 따라 큰 변동성을 보일 수 있다. Entmax는 이러한 중요한 이벤트에만 집중하여 가격 변동성을 더 정확하게 포착할 수 있게 해준다.

4. 게이팅 메커니즘과 표현 결합

4.1 어텐션 컨텍스트와 LSTM 최종 상태의 결합

이 모델은 어텐션으로 생성된 컨텍스트 벡터와 LSTM의 최종 은닉 상태를 단순히 연결하는 대신, 게이팅 메커니즘을 사용해 적응적으로 결합한다:

```
# Gating layer
self.gate = nn.Sequential(
    nn.Linear(hidden_size * 2, 1),
    nn.Sigmoid()
)

# In forward pass
combined = torch.cat([context, last_hidden], dim=1)
alpha = self.gate(combined)
fused = alpha * context + (1 - alpha) * last_hidden
```

4.2 게이팅의 이론적 중요성

이 게이팅 메커니즘은 Highway Networks와 LSTM 자체에서 영감을 받은 것으로, 모델이 컨텍스트 정보와 최종 상태 정보 사이의 균형을 데이터에 따라 적응적으로 조절할 수 있게 해준다. 이는 다음과 같은 경우에 유용하다:

1. 매우 특정한 시점(들)이 예측에 중요한 경우 → 컨텍스트에 높은 가중치
2. 전반적인 추세가 중요한 경우 → 최종 은닉 상태에 높은 가중치

5. 다단계 예측과 시퀀스 모델링

5.1 슬라이딩 윈도우 접근법

`MultiStepTimeSeriesDataset` 클래스는 슬라이딩 윈도우 방식으로 시계열 데이터를 처리한다:

```
def __init__(self, dataset, target, data_window, target_size, step,
             single_step=False):
    self.data, self.labels = [], []
```

```

start_index = data_window
end_index = len(dataset) - target_size

for i in range(start_index, end_index):
    indices = range(i - data_window, i, step)
    self.data.append(dataset[indices])

    if single_step:
        self.labels.append(target[i + target_size])
    else:
        self.labels.append(target[i:i + target_size])

```

이 접근법은:

1. 입력 윈도우(`data_window`)를 통해 과거 데이터를 관찰함
2. 출력 윈도우(`target_size`)를 통해 미래 여러 시점을 한 번에 예측함
3. 샘플링 스텝(`step`)으로 데이터 간소화 및 과적합 방지함

5.2 다단계 예측의 어려움과 해결책

다단계 예측(여러 미래 시점을 한 번에 예측)은 오차 누적 및 불확실성 증가로 인해 단일 스텝 예측보다 어렵다. 이 모델은 다음 방법으로 이를 해결한다:

1. **직접 다단계 예측**: 자기회귀적 접근법 대신, 모델이 미래 시퀀스 전체를 한 번에 예측함
2. **비선형 예측 헤드**: 2층 FC 네트워크를 통해 복잡한 비선형 패턴 포착함
3. **EntmaxAttention**: 예측에 중요한 시점에 초점을 맞추므로써 정확도 향상시킴

6. 금융 시계열 데이터에 적용된 특수 기법

6.1 변동성 관련 기술적 지표

```

def add_volatility_features(df):
    # 절대 수익률
    df['abs_return'] = df['Coffee_Price_Return'].abs()

    # 변동성 (rolling std)
    df['volatility_5d'] = df['Coffee_Price_Return'].rolling(window=5).std()
    df['volatility_10d'] = df['Coffee_Price_Return'].rolling(window=10).std()

    # 모멘텀
    df['momentum_5d'] = df['Coffee_Price'] - df['Coffee_Price'].shift(5)

    # Bollinger Band Width
    rolling_mean = df['Coffee_Price'].rolling(window=20).mean()
    rolling_std = df['Coffee_Price'].rolling(window=20).std()
    df['bollinger_width'] = (2 * rolling_std) / rolling_mean

    # Return Z-score
    df['return_zscore'] = (df['Coffee_Price_Return'] -
    df['Coffee_Price_Return'].rolling(20).mean()) / \

```

```
return df[(df['Coffee_Price_Return'].rolling(20).std() + 1e-6)
```

이론적 배경:

1. **절대 수익률(Absolute Return)**: 가격 변화의 방향과 관계없이 변동 크기 측정함
2. **롤링 표준편차(Rolling Standard Deviation)**: 특정 기간 동안의 가격 변동성 포착함
3. **모멘텀(Momentum)**: 가격 변화의 방향과 강도 측정함 (추세 추적)
4. **볼린저 밴드 폭(Bollinger Band Width)**: 시장 변동성의 상대적 수준 측정함
5. **Z-score**: 비정상적 가격 움직임 식별함 (통계적 이상치 감지)

6.2 정적 피처와 동적 피처의 결합

모델은 두 가지 유형의 입력 피처를 사용한다:

1. 동적(시간 변화) 피처:

- 거시경제 지표
- 커피 가격 자체의 과거 값 및 파생 지표
- 이러한 피처들은 시간에 따라 변화하며 LSTM이 시간적 패턴을 포착함

2. 정적(고정) 피처:

- 커피 재배 지역의 기후 데이터
- 가장 최근 수확기의 평균값을 사용함
- 이러한 피처들은 커피의 품질과 수확량에 영향을 미치는 구조적 요인 반영함

6.3 LightGBM 기반 특성 선택

코드는 LightGBM을 사용하여 정적 피처 중에서 중요한 것들만 선택하는 접근법을 언급하고 있다. 이는 다음과 같은 이점이 있다:

1. **차원 축소**: 불필요한 피처 제거로 모델 복잡도 감소시킴
2. **노이즈 제거**: 관련 없는 피처로 인한 오버피팅 방지함
3. **해석 가능성**: 중요한 기후 변수 식별 가능함

7. 손실 함수와 최적화 전략

7.1 MSE 손실

```
criterion = nn.MSELoss()
```

MSE(Mean Squared Error)는 예측값과 실제값 간 제곱 차이의 평균을 계산한다. 이 손실 함수는 큰 오차에 더 큰 페널티를 부여하므로, 이상치에 더 민감하게 반응한다.

7.2 Adam 최적화기

```
optimizer = torch.optim.Adam(model.parameters(), lr=0.001, weight_decay=1e-5)
```

Adam(Adaptive Moment Estimation)은 다음의 특성을 가진 적응적 학습률 최적화 알고리즘이다:

1. **모멘텀**: 그래디언트의 이동 평균 사용함
2. **RMSProp**: 그래디언트의 제곱에 대한 이동 평균 사용함
3. **적응적 학습률**: 파라미터별로 다른 학습률 적용함
4. **weight_decay(L2 규제화)**: 오버피팅 방지함

7.3 그래디언트 클리핑

```
torch.nn.utils.clip_grad_norm_(model.parameters(), max_norm=5)
```

그래디언트 클리핑은 그래디언트의 노름(norm)이 특정 임계값을 초과할 경우 이를 조정하는 기법이다. 이는 다음과 같은 역할을 한다:

1. **그래디언트 폭발 방지**: RNN/LSTM에서 특히 중요함
2. **학습 안정화**: 극단적인 업데이트로 인한 발산 방지함
3. **수렴 개선**: 일관된 학습 진행 보장함

8. 다양한 시계열 모델 간의 이론적 비교

프로젝트에서 언급된 다양한 모델들은 각각 다른 이론적 근거와 장단점을 가지고 있다:

8.1 VAR(벡터 자기회귀) 모델

- **이론**: 다변량 시계열의 선형 관계를 모델링함
- **장점**: 해석 가능성 높음, 변수 간 관계 명시적 표현함
- **단점**: 비선형 관계 포착 불가, 과거 데이터에 많이 의존함

8.2 LSTM 모델

- **이론**: 게이팅 메커니즘을 통한 장기 의존성 포착함
- **장점**: 비선형 패턴 학습 가능, 시퀀스 길이 제약 적음
- **단점**: 모든 과거 정보를 동등하게 압축, 모델 복잡성 높음

8.3 Attention LSTM 모델

- **이론**: 중요 시점에 집중할 수 있는 메커니즘 추가함
- **장점**: 선택적 정보 활용, 모델 해석 가능성 향상시킴
- **단점**: 일반 softmax 어텐션은 모든 입력에 가중치 부여로 평탄화 효과 발생함

8.4 Entmax Attention LSTM 모델

- **이론**: 희소 어텐션을 통한 선택적 집중 강화함
- **장점**: 중요 시점만 선택적으로 고려, 변동성 포착 능력 향상시킴
- **단점**: 모델 복잡성 증가, 하이퍼파라미터 튜닝 필요함

9. 한계점 및 이론적 개선 방향

9.1 현재 모델의 한계

1. **스무딩 효과**: 실제 가격의 급등락을 완전히 포착하지 못함
2. **기후 데이터 활용**: 단순 평균값 사용으로 정보 손실 가능성 있음
3. **손실 함수**: MSE는 방향성보다 절대적 오차에 집중함

9.2 이론적 개선 방안

1. 손실 함수 수정:

- 방향성 손실(Directional Loss): 가격 변화의 방향을 명시적으로 고려함
- 비대칭 손실(Asymmetric Loss): 상승/하락 예측 오류에 다른 가중치 부여함

2. 다양한 기후 데이터 표현:

- 평균 외에 분산, 극값, 이상치 빈도 등 추가 통계량 활용함
- 기후 패턴의 시간적 변화를 반영하는 동적 피쳐 추가함

3. 하이브리드 모델링:

- 전통적 시계열 모델과 딥러닝 모델의 앙상블 구성함
- 다양한 시간 스케일에서의 예측 결합함 (단기/중기/장기)

결론

이 Attention LSTM 모델은 커피 가격 예측을 위해 다양한 이론적 접근을 통합했다. LSTM의 장기 의존성 학습 능력, Entmax를 활용한 희소 어텐션의 선택적 집중, 그리고 금융 시계열 데이터에 특화된 피쳐 엔지니어링을 결합함으로써, 전통적인 시계열 모델보다 비선형 패턴을 더 효과적으로 포착하고자 했다. 이러한 접근법은 시계열 예측뿐만 아니라 다양한 금융 시장 예측 과제에도 확장 적용될 수 있는 이론적 기반을 제공한다.