[ZZSN]
Dokumentacja Końcowa

Piotr Frątczak
(300207)

6 czerwca 2022

1 Zadanie

Wykorzystaj modele GRU, LSTM oraz R-Transformer do modelowania danych giełdowych np. ze zbioru danych NYSE. Porównaj otrzymane wyniki. Spróbuj znaleźć istniejące rozwiązanie do modelowania giełdy i porównać jego działanie z powstałymi modelami.

2 Realizacja

2.1 Zbiór danych

New York Stock Exchange - Zbiór zawierających historyczne dane do analizy fundamentalnej i technicznej przedsiębiorstw wchodzących w skład indeksu S&P 500.
Zbiór danych składa się z czterech plików:

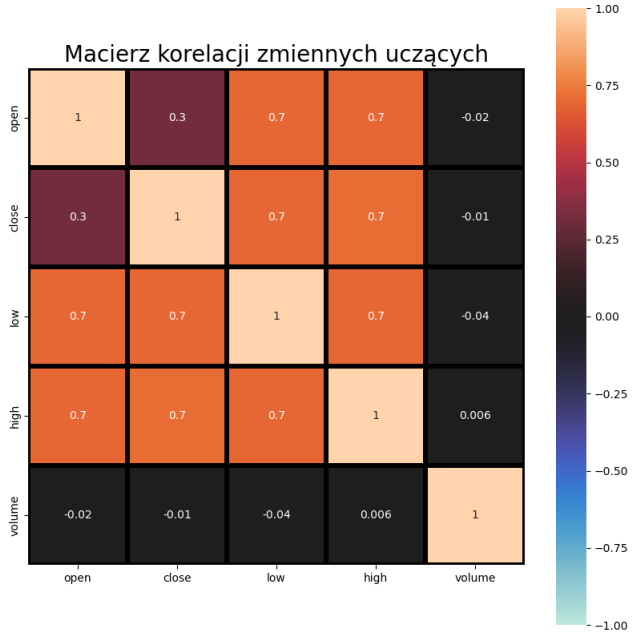
- `prices.csv` - Kursy cen akcji w okresie od 2010 do 2016. Zawierają ceny w momencie otwarcia i zamknięcia giełdy oraz ceny maksymalne i minimalne danego dnia. W danych nie wzięto pod uwagę splitów akcji (wystąpiło ok. 140 splitów akcji dla tego okresu).
- `prices-split-adjusted.csv` - Tak samo jak powyższy plik, lecz z uwzględnieniem splitów akcji.
- `securities.csv` - Ogólny opis każdego przedsiębiorstwa z podziałem na sektory.
- `fundamentals.csv` - Metryki wyciągnięte z rocznego raportu SEC 10K w latach od 2012 do 2016 wystarczające do wyciągnięcia najbardziej popularnych wskaźników analizy fundamentalnej.

Spośród dostępnych plików wykorzystano jedynie `prices-split-adjusted.csv`.

2.2 Analiza Danych

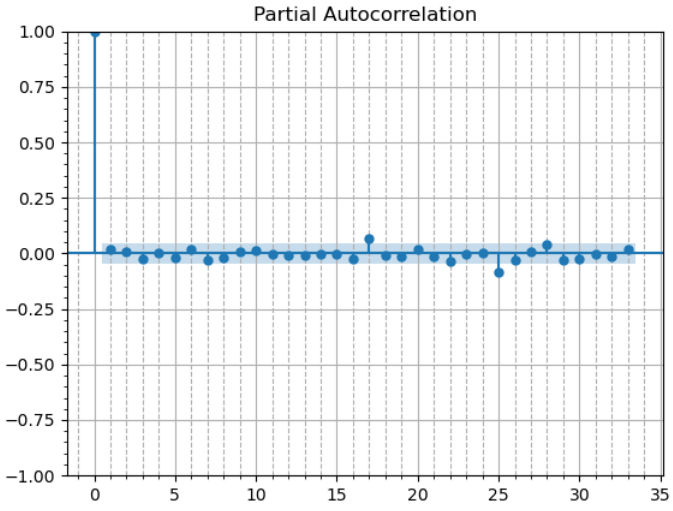
Przeprowadzono analizę danych przez obliczenie i graficzne przedstawienie korelacji między zmiennymi, aby znaleźć jak najlepsze parametry uczenia. Po przeprowadzeniu analizy okazało się również, że zbiór nie zawierał brakujących wartości i był gotowy do uczenia.

Zbadano macierz korelacji potencjalnych zmiennych uczących, aby ustalić jak duże są związki między ich wartościami.



Rysunek 1: Macierz korelacji danych uczących.

Z macierzy na Rysunku 1 wynika, że wszystkie zmienne dotyczące ceny akcji (*close* - cena zamknięcia, *open* - cena otwarcia, *high* - najwyższa cena danego dnia, *low* - najniższa cena danego dnia) poza zmienną *open* są dość silnie skorelowane ze zmienną *close* w przeciwieństwie do zmiennej *volume* - liczby akcji w obrocie. Dlatego zmienne *close, open, high i low* mają dobry potencjał do nauczenia modelu.



Rysunek 2: Wykres autokorelacji cząstkowej logarytmicznej stopy zwrotu.

Po analizie wykresu autokorelacji cząstkowej zmiennej uczonej (Rysunek 2), ustalono długości sekwencji na wartości 17 i 25. Dla wymienionych przesunięć wartości bezwzględne autokorelacji były najwyższe, pomijając przypadek oczywisty jak przesunięcie zerowe.

2.3 Zastosowane architektury

Zaimplementowane zostały modele dla poniższych architektur:

- **GRU** - Gated Recurrent Unit to sieć o podobnej, jednak uproszczonej budowie w porównaniu do LSTM, która osiąga podobne wyniki. GRU używa bramek reset i update, które decydują jakie informacje powinny zostać przekazane do wyjścia. Opiera się na idei, aby pamiętać ważne informacje nawet z dalekiej przeszłości, dopóki nie pojawią się nowe przełomowe informacje [1].
- **R-Transformer** - Recurrent Neural Network Enhanced Transformer to sieć łącząca zalety RNN i mechanizmu wielogłowej atencji przy czym pozbywa się ich wad. Dzięki RNN wykrywa zależności krótkookresowe w sekwencjach, natomiast z wykorzystaniem wielogłowej atencji wychwytuje zależności długookresowe bez wykorzystania znaczników pozycyjnych. R-Transformer osiąga znacznie lepsze wyniki niż inne architektury do przetwarzania sekwencji [2].

2.4 Założenia projektowe

- Modele przewidywały logarytmiczną stopę zwrotu w pojedynczym dniu, na podstawie danych z poprzednich dni. Modele będą realizowały zadanie regresji, polegające na przewidywaniu procentowej zmiany ceny zamknięcia, w porównaniu do poprzedniego dnia. Modele dokonywały predykcji tylko jeden dzień w przód, na podstawie danych z poprzednich dni.
- Dla każdej z cen w zbiorze danych policzono logarytmiczną stopę zwrotu symbolizującą logarytm zwrotu przy zakupie w np. w cenie zamknięcia, jeśli kupiono akcje poprzedniego dnia również w cenie zamknięcia. W taki sam sposób policzono zwroty dla cen otwarcia, najwyższych i najniższych dziennych.
- Wyniki zwrócone przez sieci uwzględniały splity akcji.
- Środowisko jest odtwarzalne, aby umożliwić weryfikację przeprowadzanych eksperymentów.
- Zbiór został podzielony na zbiór uczący, walidacyjny i testowy w proporcjach odpowiednio 80%, 10% i 10%.

2.5 Środowisko

Język programowania: Python 3.9.
Biblioteki:

- PyTorch - Implementacja LSTM i GRU, uczenie modeli.
- Pandas - Analiza i przygotowanie danych do uczenia.
- NumPy - Przetwarzanie wielowymiarowych danych jako wsparcie PyTorch i Pandas.
- Matplotlib - Wizualizacja danych, w szczególności rysowanie wykresów.
- Seaborn - Wizualizacja wyników, wsparcie Matplotlib.
- Statsmodels - Obliczanie i wizualizacja autokorelacji danych wejściowych.
- Torchmetrics - Metryki do oceny wyników wytrenowanych modeli.
- Wandb - zbieranie i agregacja wyników eksperymentów oraz przeszukiwanie parametrów.

Implementacja R-Transformera: DSE-MSU/R-transformer.

3 Eksperymenty

Eksperymenty polegały na 10-krotnym trenowaniu każdej sieci wymienionej w sekcji 2.3 na zbiorze danych z sekcji 2.1 i zbieraniu wyników predykcji na zbiorze testowym. Zbiór testowy składał się z ostatnich 10% sekwencji w zbiorze danych. Modele miały za zadanie przewidywać tylko jeden dzień w przód.

Eksperymenty zostały uruchomione 10 krotnie dla każdego przypadku z ziarnem domyślnym w projekcie równym 1111.

3.1 Model bazowy

Model bazowy jest poziomem odniesienia, służy do porównania czy wytrenowanie modeli w ogóle ma sens. W tym wypadku model bazowy to podejście naiwne polegające na zwróceniu wartości logarytmicznej stopy zwrotu z dnia poprzedniego.

3.2 Wyniki

Do porównania modeli użyto metryk: Root-mean-square error (RMSE), Mean absolute percentage error (MAPE), Skorygowany R^2 oraz stratę na zbiorze testowym.

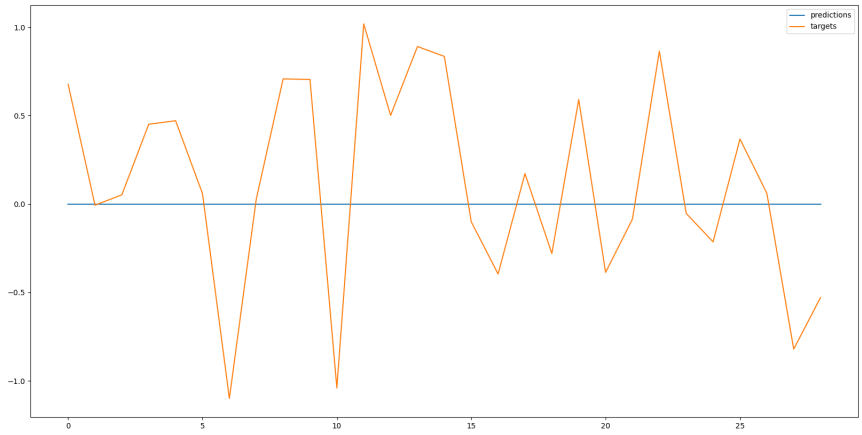
Skorygowany R^2 jest skorygowaną miarą dopasowania modelu. Identyfikuje procent wariancji w zmiennej przewidywanej wyjaśniony przez dane wejściowe lub dane wejściowe. R^2 ma tendencję do optymistycznego szacowania dopasowania regresji liniowej. Zawsze wzrasta wraz z liczbą efektów zawartych w modelu. Skorygowany R^2 próbuje skorygować to przeszacowanie. Skorygowany R^2 może się zmniejszyć, jeśli określony efekt nie poprawi modelu.
Skorygowane R^2 jest zawsze mniejsze lub równe R^2. Wartość 1 wskazuje model, który doskonale przewiduje wartości w zmiennej przewidywanej. Wartość mniejsza lub równa 0 wskazuje model, który nie ma wartości predykcyjnej. W rzeczywistym świecie skorygowany R^2 znajduje się między tymi wartościami.

Wyniki w Tablicy 3.2 posortowano rosnąco według skorygowanego R^2.

Model	dł. sekwencji	Adjusted R^2	MAPE	RMSE	Test MSE loss
naiwny	nd.	-0.98130	13.4717	1.3434	0.0282
RT	25	-0.09558	9.0073	0.9990	0.0078
RT	17	-0.01834	3.2292	0.9584	0.0072
GRU	25	-0.00005	1.0200	0.9545	0.0071
GRU	17	**-0.00002**	**1.0049**	**0.9498**	**0.0070**

Tablica 1: Miary skuteczności modeli.

Według otrzymanych wyników, najlepiej sprawdził się model GRU dla krótszej sekwencji (17 dni wstecz) oraz wszystkie przetestowane modele uzyskały wyniki lepsze niż dla podejścia naiwnego. Jednak po weryfikacji graficznej otrzymanych wyników okazuje się, że modele nie sprawdziły się dobrze.
Model nauczył się przewidywać stałą wartość jak widać na Rysunku 3. W ten sposób ograniczył błąd wychwytywany przez każdą z metryk. Dane zostały ustandaryzowane, dlatego są skupione wokół zera, to spowodowało, że proste podejście otrzymało najlepsze wyniki.



Rysunek 3: Predykcje modelu GRU dla sekwencji o dł. 17.

3.3 Wnioski

Wyniki nie są zadowalające. Zupełnie nie udało się wytrenować działających modeli. Zawiodło wyszukiwanie hiperparametrów.

Problemem okazał się szczególnie fakt, że R-Transformer ma wiele hiperparametrów, przez co należy zbadać bardzo szeroką gamę możliwości, aby mieć szansę trafić na dobre ustawienia. Spędzono więcej czasu na poszukiwanie hiperparametrów R-Transformera, ponieważ spodziewano się, że osiągnie lepsze wyniki niż prostsza architektura GRU.

Koniecznym w tej sytuacji jest dalsze przeszukiwanie przestrzeni hiperparametrów w celu znalezienia działającej konfiguracji. Korzystne będzie oparcie się o wiedzę ekspercką dotyczącą architektur oraz dziedziny przewidywania giełdy.

Pozytywnym aspektem wykonania projektu jest gotowa platforma do poszukiwania hiperparametrów dla R-Transformera oraz GRU przy wykorzystaniu Wandb.

4 Zmiany względem dokumentacji wstępnej

4.1 Zmniejszenie zakresu projektu

Z powodu zmniejszenia zespołu projektowego z dwóch osób do jednej, ograniczono zakres projektu. Z tego powodu wykluczono z rozważań architekturę LSTM (zmiana w sekcji 2.3) oraz ograniczono liczbę wariantów t z trzech do dwóch (zmiana w sekcji 3). t oznacza długość sekwencji, na podstawie której wykonywane są predykcje.

4.2 Modyfikacja elementów eksperymentów

Nie użyto miary Mallows' Cp ze względu na niewygodę implementacji. Zmiana w sekcji 3.

Nie porównano wyników z istniejącym rozwiązaniem, ponieważ osiągnięte wyniki nie byłyby konkurencyjne z żadnym zadowalającym rozwiązaniem.

4.3 Modyfikacja założeń projektu

Po weryfikacji założeń przewidywania ceny zamknięcia akcji, zmieniono cel przewidywań na logarytmiczną stopę zwrotu. Logarytmiczna stopa zwrotu jest reprezentacją używaną przez profesjonalistów na giełdzie dzięki praktyczności w obliczeniach zwrotu z inwestycji. Użycie logarytmu powoduje, że operacja dzielenia może być zastąpiona odejmowaniem. Ponadto jego zaletą jest fakt, że jego wartości skupiają się wokół zera, co ułatwia standaryzację danych i zwiększenie wydajności uczenia. Modyfikację wprowadzono w sekcji 2.4.

Użyto zbioru danych z uwzględnionymi splitami akcji i jednocześnie uwzględniano je w wynikach. W przeciwnym przypadku, ceny ulegały gwałtownych niewytłumaczalnych z zewnątrz zmian. Modyfikacja również w sekcji 2.4.

4.4 Uaktualnienie środowiska

Zmieniono wersję języka programowania z Python 3.8 na Python 3.9, aby iść z duchem rozwoju technologii i zapewnić dłuższe wsparcie dla projektu. Do tego rozszerzono listę użytych zależności wraz z implementacją. Zmiana dotyczy sekcji 2.5.

5 Napotkane problemy

Po realizacji dokumentacji wstępnej zespół zmniejszył się z dwuosobowego do jednoosobowego. Kwestia została rozwiązana przez zmniejszenie zakresu projektu zgodnie z punktem w podsekcji 4.1. Jednak nie sprawiło to, że projekt był o połowę mniej wymagający, co spowodowało większe ograniczenia czasowe, które wpłynęły szczególnie na możliwości wykonania eksperymentów.

Napotkano na wiele trudności z uruchomieniem szkieletu projektu zaproponowanego w zasadach realizacji projektu (2021L-ZZSN/template/). Po licznych próbach postanowiono zaniechać użycia szkieletu.

Trudności z odpowiednim znalezieniu odpowiednich nastaw hiperparametrów. Próbowano wykonać wiele przeglądów hiperparametrów, jednak nie natrafiono na zadowolające rozwiązanie.

6 Kod źródłowy

Kod źródłowy jest dostępny w repozytorium: piotrfratczak/zzsn-nyse.

Bibliografia

[1] Kyunghyun Cho i in. „Learning phrase representations using RNN encoder-decoder for statistical machine translation". W: *arXiv preprint arXiv:1406.1078* (2014).

[2] Zhiwei Wang i in. „R-transformer: Recurrent neural network enhanced transformer". W: *arXiv preprint arXiv:1907.05572* (2019).