Някои бележки по прогнозирането на фондовите пазари

Въведение

Как да се предвиди цената на акциите, все още е горещ изследователски проблем за инвеститорите и изследователите във финансовата област. Прогнозирането на цените на акциите се превръща в изключително предизвикателна задача поради високия шум, нелинейността и нестабилността на данните от времевите редове на цената на акциите. Основната идея на всички усилия в тази насока е хипотезата за ефективния пазар, която казва, че цената на акциите може да бъде предвидена от историческите данни за търговията. Специалистите все по-често поглеждат към широкия набор съвременни възможности предоставяни от дълбокото машинно обучение и все по-успешно прилагат методите му. Повечето модели ползват разточителен набор от данни, включващи данни за търговията на конкретната акция, данни за индекси, валути, финансово-счетоводна информация, данни за макроикономически показатели и.н. Част от моделите се обучават и върху данни свързани със сентимента на пазара, като новини, туитове, постове в социални мрежи. Има конструирани и сентимент индикатори, които могат да бъдат ползвани директно в съвременните невронни мрежи.

Целта на този проект е да се анализира ефективността на различни модели за прогнозиране върху данни от фондовия пазар и да се предскаже бъдещата цена на дадена акция.

Методология

1. Данни

Моделите биват захранван с два вида данни, оформени в два главни сета (***stocks_df.csv*** и ***company_dataset_ts.csv***).

Company_dataset_ts.csv съдържат данните (под формата на времеви редове с 30 дневни прозорци на цената на затваряне) за конкретна компания, както и седем изчислени индикатори. В ***stocks_df.csv*** са събрани данни за 85 други характеристики, включващи индекси, валути, инфлационни индикатори, криптовалути, съкровищни бондове, ETFs, взаимни фондове, както и данни за други 50 компании от S&P 500. Пълен списък с ползваните данни, както и източниците на данни са предоставени в ***get_data\data\indicators_list.csv***. Основния източник на данни е Yahoo Finance. Данните са подложени на задълбочен анализ, като допълнително е изготвен анализ за корелации, хетероскедастичност, серийни корелации и др. Направена е оценка на важните характеристики посредством PCA и XGBoosting, която по-късно не беше приложена, предвид оскъдното количество данни. Подбрана е извадка от 2267 наблюдения обхващащи периода от 2013 до 2021 г. Данните са разделени на трениращи и тестови в съотношение 85%:15% (1926, 341). Част от трениращите данни (около 20% - 22%) се ползват за валидация.

Основни моменти от събирането, анализа, визуализацията и предподготовката на данните може да се намерят в ***get_data\SMP_GateringData.ipynb*** и ***get_data\ SMP_Data_Analysis.ipynb***.

Част от предподготовката на данните е изнесена в основните файлове (***Stock_Market_Price_Predictor_Base_Models.ipynb*** и ***Stock_Market_Price_Predictor_Hybrid_Models.ipynb*** в главната директория на проекта). Данните за трениране са размесени на случаен принцип[1]. Данните за тестване следват времевата си хронология, за да може моделите да бъдат оценени максимално близо до реалните работни условия. Следва min-max нормализация на данните и преоразмеряване, подходящо за вход в невронните архитектури.

2. Основни архитектури

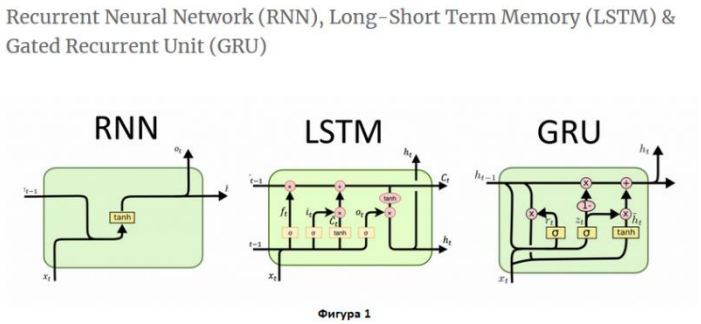
Преди конструирането на хибридните модели са тествани няколко архитетури върху различните канали от данни. Основното внимание е насочено към LSTM и GRU, като по-съвремения и подобрен вариант на рекурентна невронна архитектура.

Рекурентните невронни мрежи решават проблема със „забравянето“ на информацията. Изходите на невронните елементи на следващите слоеве имат синаптични връзки с невроните на предишните слоеве. Това води до възможността за отчитане на

[1] *Фондовите пазари са характерни с оформянето на дългосрочни трендове. Избирането на подход без размесване на времевите серии не доведе до задоволителни резултати, тъй като Ковид кризата през последните години изкриви пазарите в голяма степен. Размесването на трениращите данни донякъде не позволи на моделите да заучат предходните по-стационарни трендове.*

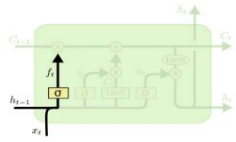
резултатите от преобразуването на информацията от невронната мрежа на предходния етап за обработка на входния вектор на следващия етап от функционирането на мрежата.

Биологичната интелигентност обработва информацията постепенно, като същевременно поддържа вътрешен модел на това, което обработва, изграден от минала информация и непрекъснато актуализиран при постъпване на нова информация. Повтарящата се невронна мрежа (RNN) приема същия принцип, макар и в изключително опростена версия: тя обработва последователности чрез итериране през елементите на последователността и поддържа състояние, което съдържа информация относно това, което е видяла досега. Всъщност RNN е вид невронна мрежа, която има вътрешен цикъл (Фигура 1). Състоянието на RNN се нулира между обработката на две различни, независими последователности (примерно две проби в партида), така че все още смятате една последователност за една точка от данни: един вход към мрежата. Това, което се променя е, че тази точка от данни вече не се обработва в една стъпка, а по-скоро мрежата вътрешно обикаля елементи на последователност.



Фигура 1

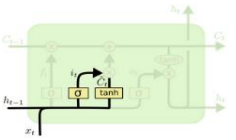
Простите RNN имат сериозен проблем: въпреки че теоретично би трябвало да може да се запази в момент t информацията от предходните времеви стъпки, подобни по-дългосрочни зависимости се оказват невъзможни за научаване на практика. Това се дължи на проблема с изчезващия градиент, ефект, който е подобен на това, което се наблюдава при неповтарящи се мрежи, които са много слоеве дълбоки. LSTM и GRU, които са предназначени да решат тези проблеми.



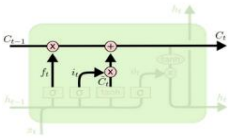
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LTSM

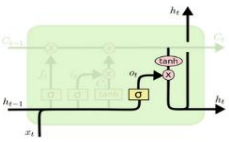
Мрежите за дългосрочна памет LSTM, са специален вид RNN, способни да учат по-дългосрочни зависимости. При тях, в повтарящият се модул , вместо да има един слой (than в RNN), има четири слоя, всеки с различно предназначение (Фигура 1). Основна ключова роля за този тип клетки е остта t-1 → C_t, чието предназначение е да съхрани част от състоянието на клетката и да го пренесе непроменено до следващата. Сигмоидният слой извежда числа между нула и едно, описвайки колко от всеки компонент трябва да бъде пропуснат. LSTM има три от тези порти, за да защитава и контролира състоянието на клетката. Разгледана на стъпки, математиката, който стои зад всяка от тях е:



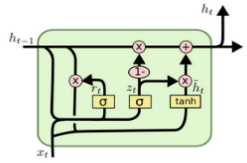
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



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



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

GRU

GRU е по-опростена и рационализирана версия на архитектурата LSTM. При този вид клетка вътрешните слоевете са намалени на три, а гейтовете за вход и забравяне са съчетани в единен гейт за „актуализиране“. GRU става все по-популярен.

Bidirectional RNN

Двупосочният RNN е често срещан вариант на RNN, който може да предложи по-голяма производителност от обикновения RNN при определени задачи. Двупосочният RNN използва две редовни RNN, като слоевете GRU и LSTM. Чрез обработка на последователностите и в двете посоки, двупосочния RNN може да улови модели, които могат да бъдат пренебрегнати от

еднопосочния RNN. От особена важност при работа с двупосочни модели за решаване на задачи за прогнозиране е да не се допуска обучение с валидационни данни, тъй като архитекрурата им предполага „надникване в бъдещето “. За тази цел е по-подходящо те да бъдат ползвани в енкодер/декодер архитектури за генериране на изходния вектор на енкодера. Този тип архитектура беше тествана в проекта предимно от любопитство.

3. Метрики

При решаване на регресионни задачи обичайната мярка за оценка на загубата е средната квадратична грешка (MSE), а не средната абсолютна грешка (MAE), защото за разлика от MAE, тя е гладка около нулата и като такава има дефинирана производна там. Това е от съществено значение при изчисляване на градиента.

По-ниската стойност на MAE и MSE предполага по-висока точност на регресионния модел. MAE е по-стабилен към данни с отклонения. В проекта MAE е наблюдавана, чрез добавяне като метрика в compile().

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

Фигура 2

Експериментални резултати

1. Модели

Имплементацията на моделите може да бъде разгледан в Stock_Market_Price_Predictor_Base_Models.ipynb и Stock_Market_Price_Predictor_Hybrid_Models.ipynb в главната директория на проекта!

В проекта са разгледани няколко базови архитектури от типа Many-to-One базирани на двата входни сета. На база анализите на тези модели са предложни два хибридни модела с два входни клона, по които влизат двата типа данни, след което се обединяват. Очакванията бяха хибридните модели да подобрят резултатите от индивидуалните модели. В долната таблица (Таблица 1) са систематизирани архитектурите на моделите:

Таблица 1

Model	Type		Dataset	Branches	Layers	Optimizer
BiLSTM	Many-to-One	Simple	stocks_df.csv	1	2-Bidirectional 3-Dense	RMSprop
GRU	Many-to-One	Simple	stocks_df.csv	1	2-GRU 3-Dense	Adam
Base Model	Many-to-One	Simple	company_dataset_ts.csv	1	4-Dense	RMSprop
LSTM	Many-to-One	Simple	company_dataset_ts.csv	1	6-LSTM 2-Dense	RMSprop
GRU	Many-to-One	Simple	company_dataset_ts.csv	1	6-GRU 2-Dense	RMSprop
LSTM-GRU	Many-to-One	Hybrid	stocks_df.csv company_dataset_ts.csv	2	b1: 6-LSTM + 2 Dense b2: 2-GRU + 3-Dense concatenated + 3-Dense	RMSprop
LSTM-BiLSTM	Many-to-One	Hybrid	stocks_df.csv company_dataset_ts.csv	2	b1: 6-LSTM + 2 Dense b2: 2-BiLSTM + 3-Dense concatenated + 3-Dense	RMSprop

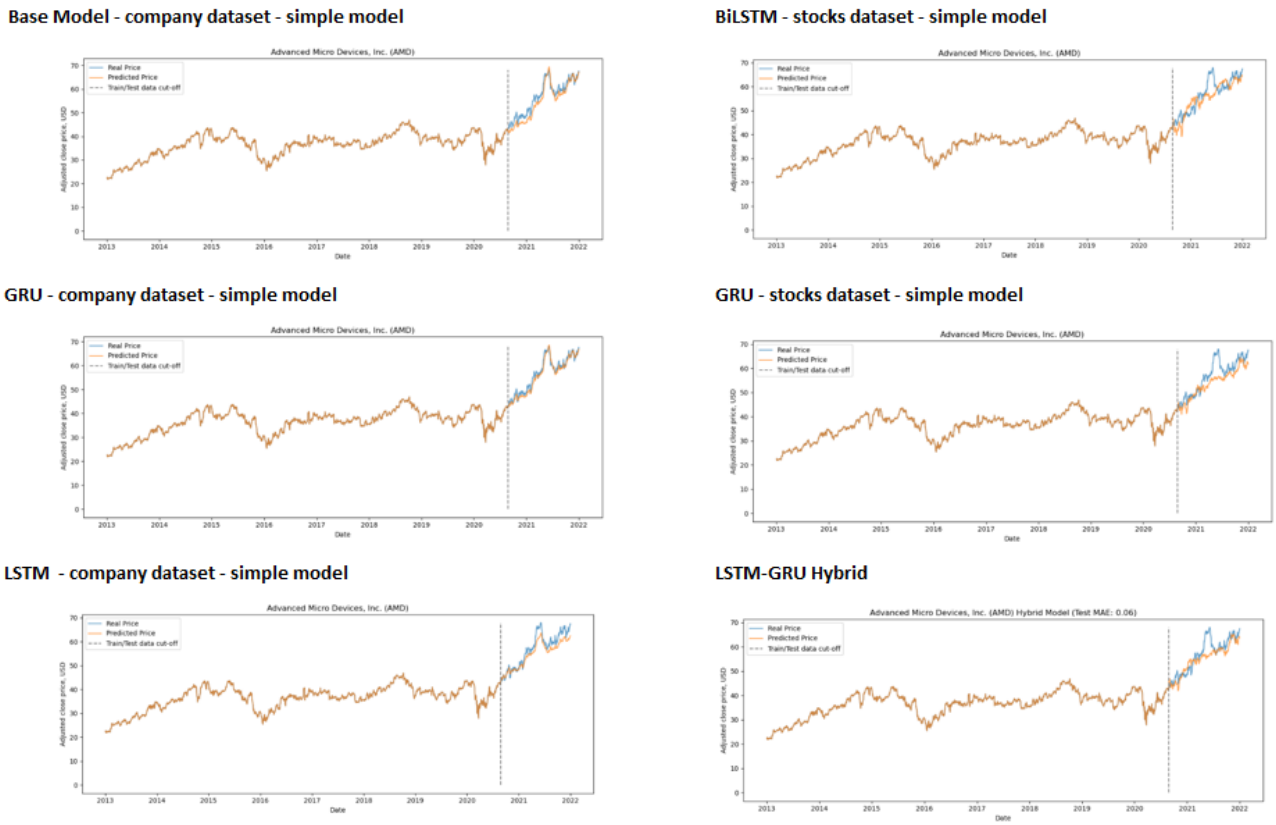
2. Резултати и анализи

След тестване моделите отчетоха следните резултати (Таблица 2):

Таблица 2

Model	MAE	Epoch	Batch Size	Validation Split
BiLSTM	Test MAE: 0.08	90	128	-
GRU	Test MAE: 0.07	150	128	0.225
Base Model	Test MAE: 0.10	120	128	0.25
LSTM	Test MAE: 0.05	70	128	0.225
GRU	Test MAE: 0.04	65	128	0.225
LSTM-GRU	Test MAE: 0.05	150	128	0.225
LSTM-BiLSTM	Test MAE: 0.06	150	128	-

На графиките (Фигура 3) са представени примерни прогнози генерирани от моделите:



Фигура 3

Заключение

От направените анализи може да се направи заключение, че простите модели, работещи директно с данните свързани с акцията се представят по-добре. Сателитните данни добавени в хибридният модел показват тенденция за доминация и влошават прогнозните резултати. Това може да се дължи на неправилната пропорция между данните или на недостатъчно прецизна архитектура/тренировка на невронната мрежа. По-нататъшната работа по проекта ще бъде свързана с тестването на още няколко хибридни архитектури, които да дават приоритет на данните генерирани от самата акция. За целта могат да бъдат потърсени комбинации на характеристики от stocks_df.csv, при които да има случаен подбор на характеристики или да се пробват получените след анализа по-важни характеристики. Могат да бъде разширено разнообразието на сателитните характеристики с данни за пазарен сентимент, новини, туитове и т.н. Такива тестове бяха направени с включване на сентимент индикатор на туитове (TSI), но за

съжаление данните за него са от 2018 година и резултатите на моделите, в които беше включен не бяха представителни. Особено значими биха били данните от счетоводните отчети на компаниите. На настоящият етап от проекта това не беше възможно поради трудният и платен достъп по специфичен тип информация.

Ползвани източници

1. *Deep Learning with Python, Second Edition, François Chollet*
2. *https://finance.yahoo.com/*
3. *http://colah.github.io/posts/2015-08-Understanding-LSTMs/*
4. *http://dprogrammer.org/rnn-lstm-gru*
5. *https://keras.io/api*
6. *https://www.tensorflow.org*
7. *google.com, github.com, stackoverflow.com (irreplaceable!)*