

پیش بینی قیمت سهام با استفاده از شبکه های بازگشتی

(۱) چکیده

در این مقاله، ما مدل های تجزیه و تحلیل چند متغیره را برای پیش بینی حرکت قیمت سهام در داده های سهام بانک صادرات پیاده سازی کردیم. قیمت سهام به عوامل مختلف و پویایی پیچیده آنها بستگی دارد که آنها را به یک مسئله دشوار در دنیای واقعی تبدیل می کند. هدف از این مقاله تجزیه و تحلیل توانایی یک شبکه عصبی برای حل موثر این مشکل است. شبکه های عصبی بازگشتی (RNN) توانایی خود را در رسیدگی به مشکلات پیچیده سری زمانی نشان داده اند. مدل های مختلف چند متغیره را بر اساس معماری های مختلف RNN مانند GRU و LSTM مورد تجزیه و تحلیل قرار داده و آنها را با یکدیگر مقایسه می کنیم. از روش محاسبات مبتنی بر RNN استفاده کرده ایم و مدل هایی برای یافتن وابستگی های زمانی و پیش بینی ارزش سهام یک شرکت خاص از تاریخ گذشته سهام آن استفاده شده است. یک معماری شبکه عصبی چند متغیره را در پیش بینی قیمت سهام پیشنهاد می دهیم و خطای پیش بینی این مدل ها را با هم مقایسه می کنیم. از نتایج، استنباط می کنیم که با توجه به تعداد داده ها مدل GRU پیش بینی بهتری نسبت به LSTM دارد. اما برای داده های بیشتر، با توجه به اینکه مدل GRU پیچیدگی کمتری نسبت به LSTM دارد؛ LSTM بهتر عمل میکند.

(۲) مقدمه

پیش بینی قیمت سهام یکی از مهمترین مشکلات تجاری است که توجه همه سهامداران را به خود جلب کرده است. از اطمینان پیش بینی و پیچیدگی الگوریتم های مورد استفاده در روند حل این مشکل برای بهبود عملکرد استفاده می

کنیم. با این حال، روشهایی که ما هنوز یافته ایم یا بر اساس فرضیات رگرسیون خطی ساده هستند (مانند ARIMA) یا از داده های موجود استفاده کامل نمی کنند و هنگام پیش بینی فقط یک فاکتور را در نظر میگیرند (مدلهای تک متغیر غیر خطی مانند TAR, ARCH و مدل های یادگیری عمیق). برخی از محققان همچنین ترکیبی از ANN و منطق فازی را برای استفاده از استدلال انسان مانند این مسئله امتحان کرده اند. اما پیش بینی سهام هنوز روش مشخصی ندارد. قیمت سهام بسیار پویا است و روابط غیر خطی دارد و همزمان به عوامل زیادی وابسته است. ما سعی می کنیم با استفاده از تحلیل چند متغیره این مشکل پیش بینی بازار سهام را برطرف کنیم.

از آنجا که سری های زمانی چند متغیره ویژگی های بیشتری نسبت به سری های زمانی تک متغیره دارند، بیشتر آموزنده هستند، بنابراین تقریباً همیشه بهتر است از مدل چند متغیره برای پیش بینی روند سیستم های پیچیده مانند سهام استفاده شود. شبکه های عصبی بازگشتی (RNN) و زیرمجموعه های آن مانند GRU و LSTM در داده های متوالی دیگر مانند امواج صوتی، تغییرات سری زمانی و پردازش زبان طبیعی عملکرد خوبی را نشان داده است.

از روش های مختلف یادگیری عمیق، یعنی GRU و LSTM برای مدل سازی مسئله خود استفاده کرده ایم. ثابت شده است که الگوریتم های یادگیری عمیق توانایی شناسایی الگوهای موجود در داده ها و بهره برداری از آنها را با استفاده از یک فرایند یادگیری دارند. برخلاف سایر الگوریتم های آماری و یادگیری ماشین، معماری های یادگیری عمیق قادر به یافتن وابستگی های کوتاه مدت و همچنین طولانی مدت در داده ها و پیش بینی های خوب با یافتن این روابط پنهان هستند.

ما روش کار خود را در ۳ مرحله ارائه داده ایم. ابتدا داده ها را پیش پردازش می کنیم تا داده ها چند بعدی شوند و برای ساختارهای شبکه ما مناسب باشند. بعد، ما داده ها را به مجموعه های آموزش (train) و آزمون (test) تقسیم می کنیم و مدل های خود را بر روی داده های آموزش، آموزش می دهیم. در مرحله آخر، ما با استفاده از مدل های آموزش داده شده در مرحله قبل در مورد داده های آزمون، پیش بینی می کنیم و ماتریس های خطای مختلف را محاسبه و تحلیل می کنیم. این مقاله در چهار قسمت بررسی است. بخش (۳) مفاهیم مربوط به معماری های مختلف مورد استفاده را ارائه می دهد، قسمت (۴) روش مورد استفاده برای انجام این آزمایش را نشان می دهد، قسمت (۵) حاوی نتایجی است که به دست آوردیم و نتیجه گیری در قسمت (۶) انجام می شود.

(۳) مرور مفاهیم

در این مقاله از معماری های زیر استفاده شده است:

شبکه های عصبی بازگشتی (RNN):

RNN دسته ای از شبکه های عصبی است که در آن خروجی مرحله قبل همراه با ورودی عادی به عنوان ورودی به مرحله فعلی تغذیه (feed) می شود. در ANN های feed forward ، تمام ورودی ها و خروجی ها از یکدیگر مستقل هستند، اما در مواردی مانند پیش بینی سری زمانی ، مقادیر قبلی مورد نیاز است و بنابراین نیاز به یادآوری مقادیر قبلی است. مشخص شده است که RNN از مشکل گرادیان ناپدید شده ((vanishing gradient رنج می برد. همانطور که خطا را از طریق شبکه منتشر می کنیم ، باید از حلقه زمانی عبور کند - لایه های پنهان با استفاده از وزنه برداری به موقع به خود متصل می شوند. از آنجا که این وزن بارها و بارها بر روی خودش اعمال می شود ، باعث می شود که گرادیان به سرعت کاهش یابد. در نتیجه ، وزن لایه ها در انتهای سمت چپ بسیار کندتر از وزن لایه های سمت راست به روز میشوند. این یک اثر دومینو ایجاد می کند زیرا وزن لایه های سمت چپ چپ ورودی های لایه های راست را تعریف می کند. بنابراین ، کل آموزش شبکه آسیب می بیند ، و به آن مشکل گرادیان محو ((vanishing gradient می گویند.

حافظه کوتاه مدت بلند مدت (LSTM):

LSTM یک شبکه RNN است که توسط Jürgen Schmidhuber و Sepp Hoch Reiter در سال 1997 برای حل مشکل vanishing gradient در RNN ها پیشنهاد شده است. LSTM از گیت های زیر برای حل مشکل استفاده می کند:

Forget Gate: اگر true تنظیم شود ، سلول اطلاعات به دست آمده از لایه های قبلی را فراموش می کند.

Input Gate: انتخاب می کند که کدام مقدار از ورودی وضعیت حافظه را به روز کند.

Output Gate: بر اساس ورودی و حافظه سلول ، خروجی سلول را انتخاب می کند.

(GRU) Gated Recurrent Unit:

نوعی از RNN است که توسط Kyunghyun Cho و همکارانش در سال 2014 معرفی شده است. مانند یک واحد LSTM بدون درگاه خروجی است. پارامترهای کمتری نسبت به LSTM دارد و از پیچیدگی کمتری برخوردار است. GRU عملکرد بهتری نسبت به LSTM در مجموعه داده های کوچکتر نشان داده است، اما در کل نسبت به LSTM ضعیف تر است.

۴) روش انجام کار

داده های خام:

ما از قیمت های سهام بانک صادرات که از سایت شرکت مدیریت فناوری بورس تهران (TSETMC) و داده های مربوط به شاخص ۵۰ شرکت برتر، شاخص صنعت، شاخص کل و RSI (که یک اندیکاتور در تحلیل تکنیکال است) استفاده می کنیم. داده های ما شامل 1631 رکورد از قیمت روز سهام از تاریخ 1389-01-07 تا 1397-12-12 است. هر رکورد حاوی اطلاعاتی از قیمت پایانی، اولین قیمت، بیشترین قیمت، کمترین قیمت، آخرین معامله، قیمت دیروز، تعداد معاملات، حجم معاملات، ارزش معاملات، شاخص ها و اندیکاتور مذکور در آن روز است.

پیش پردازش داده ها:

ابتدا برخی از داده های زائد را حذف می کنیم و برای یکسان سازی دامنه داده ها، نرمال سازی Min-Max را اعمال کردیم و مقادیر را در محدوده 0 تا 1 نرمال کردیم.

این داده ها به داده های آموزش (train) ، اعتبارسنجی (valid) و آزمون (test) تقسیم شدند. داده های آموزش شامل سوابق از 1389-01-07 تا 1393-12-27 ، داده های اعتبارسنجی شامل سوابق از 1394-01-05 تا 1395-04-29 و داده های آزمون شامل سوابق از 1396-12-23 تا 1397-12-19 است.

انتخاب ویژگی ها:

الگوریتم های مختلفی برای انتخاب ویژگی های کاربردی برای استفاده در شبکه های عصبی وجود دارد. ما در این مسئله از الگوریتم رگرسیون قدم به قدم برای انتخاب ویژگی ها استفاده کردیم. و نتیجه اجرای الگوریتم لیستی از ویژگی های آخرین قیمت، قیمت دیروز، کمترین قیمت، بیشترین قیمت، ارزش معاملات، حجم معاملات، شاخص ۵۰ شرکت برتر و شاخص کل است. که بیشترین همبستگی را با قیمت پایانی دارند.

روند آموزش:

داده ها را در دو معماری یادگیری عمیق GRU و LSTM برای تحقیقات خود آموزش می دهیم. RNN نوع خاصی از شبکه عصبی است که در آن ارتباطات به صورت دایره مستقیم بین واحدهای محاسباتی برقرار می شود. RNN برخلاف شبکه های عصبی پیش رو (feed-forward) ، از حافظه داخلی برای یادگیری از توالی دلخواه استفاده می کند. هر واحد در RNN دارای یک تابع فعال سازی و وزن است. تابع فعال سازی متغیر است و ارزش واقعی دارد. وزن ها قابل اصلاح هستند. GRU و LSTM زیرمجموعه های معماری RNN هستند. هر شبکه ای که ایجاد کرده ایم از سه لایه سلول RNN مربوطه و یک لایه Dense با یک سلول در انتها استفاده می کند.

تست و محاسبه خطا:

هر مدل روی مجموعه آزمون آزمایش شده است و میانگین مربعات خطا (MSE) ، میانگین ریشه مربع خطا (RMSE) و R2-SCORE آنها محاسبه شده است.

الف) مدل چند متغیره LSTM:

پارامترهای مختلف ANN به شرح زیر است:

مراحل زمانی: 20

سلول های عصبی در هر لایه: 40 و 35

میزان یادگیری: 0.01

اندازه دسته: 64

کل پارامترهای قابل آموزش: 19360

داده های آموزش به این شبکه تغذیه می شود و مدل برای 500 دوره بر روی داده های آموزش، آموزش داده می شود و توسط داده های اعتبارسنجی تایید می شود.

ب) مدل چند متغیره GRU:

پارامترهای مختلف این ANN به شرح زیر است:

مراحل زمانی: 40

سلول های عصبی در هر لایه: 40 و 35

میزان یادگیری: 0.001

اندازه دسته: 64

کل پارامترهای قابل آموزش: 14781

داده های آموزش به این شبکه داده می شود و مدل برای 500 دوره بر روی داده های آموزش آموزش داده می شود و با داده های اعتبارسنجی اعتبار می یابد.

۵) ابزار و فناوری مورد استفاده

از زبان برنامه نویسی پایتون برای این پروژه استفاده کردیم. از فریمورک تنسورفلو و Keras که یک API شبکه عصبی سطح بالا است و در پایتون نوشته شده است. برای محیط توسعه ما از Jupyter Notebook استفاده کردیم. از Plotly و Matplotlib برای تجسم داده ها ، از Numpy برای عملیات آرایه های مختلف و از Pandas برای تجزیه و تحلیل داده ها استفاده کردیم.

۶) نتایج

این آزمایشات برای دو مدل یادگیری عمیق که آموزش داده ایم انجام شده است. نتایج نهایی بدست آمده در جدول 1 زیر نشان داده شده است.
** جدول 1: نتایج مدل های مختلف بر روی داده های آزمون **

مدل	MSE	RMSE	R2-SCORE
GRU	0.00064159547	0.02532973494	0.96062510795
LSTM	0.00140960651	0.03754472689	0.92937548059

نتایج حاصل از جدول نشان داد که با توجه به تعداد داده ها مدل GRU عملکرد بهتری نسبت به مدل LSTM دارد و از پیچیدگی کمتری نیز برخوردار است.

۷) نتیجه گیری

ما برای حل مشکل قیمت سهام ، رویکرد شبکه عصبی چند متغیره را پیشنهاد داده ایم. نتیجه میگیریم که مدل های ANN چند متغیره به وضوح توانایی پیش بینی بهتری نسبت به مدل های دیگر دارند. و از داده ها استفاده بهتری میکنند. ما یک رویکرد مبتنی بر یادگیری عمیق چند متغیره را برای پیش بینی قیمت سهام پیشنهاد کردیم. رویکردی که ما پیشنهاد کردیم تنها پس از مقایسه با سایر روش های پیش بینی سهام قابل تحکیم و استناد است.