

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه پیام نور استان تهران

مرکز تهران شمال

گروه مهندسی کامپیوتر

پروژه کارشناسی

رشته ی کامپیوتر

گرایش نرم افزار

عنوان پروژه :

افزایش احتمالات در بورس با استفاده از هوش مصنوعی

استاد راهنما :

استاد رضوی ابراهیمی

تهیه کننده :

محمد احمدیان هنرمند

دی ماه ۱۳۹۹

کلیه حقوق مادی مترتب بر نتایج مطالعات، ابتکارات
و نوآوری های ناشی از این پروژه متعلق به
دانشگاه پیام نور تهران / مرکز تهران شمال
می باشد.

چکیده:

بزرگترین قطب اقتصادی در بزرگترین کشورهای اقتصادی، بورس میباشد. امروزه قدرت اقتصادی کشورها را با ارزش بورس آن کشورها میسنجند. بزرگترین بورس های جهان، پرمعامله ترین و پر حجم ترین بازارهای مالی جهان هستند. از آنجا که حجم معاملات در این بازارها بسیار بالاست، پس تعداد معامله کنندگان و تریدرها نیز در بازارها بسیار زیاد میباشد. از طرفی چون تعداد افراد و حجم معاملات بالاست، میتوانیم با قدرت هوش مصنوعی تا حدودی رفتار بازار را پیش بینی کنیم. هوش مصنوعی روز به روز در بازارهای مالی دنیا فراگیرتر میشود به طوری که تمام کارگزاری ها (بروکر) بلااستثنا از دانشمندان این حوزه برای به روز بودن در صنعت خود، استفاده میکنند. اما پیش بینی بازار کار ساده ای نیست. برای این کار هم باید در بازار های مالی بسیار نخبه بود و هم در علم هوش مصنوعی. در علم اقتصاد باید بدانیم چه دیتایی برای پیش بینی بازار لازم و مفید است و به عبارتی بازار به آن شاخص وابسته است [۲]، در علم هوش مصنوعی نیز، لازم است که بدانیم چه شبکه هایی با چه تنظیماتی میتواند آن دیتای ما را به خوبی روی مدل آموزش دهد. کار ساده ای نیست به طوری که بهترین شبکه ای که دانشمندان این حوزه ساخته اند، تنها میتواند با احتمال ۶۰ درصد بازار را به خوبی پیش بینی کند. بهترین دیتا و بهترین شبکه ها در این زمینه را فراخواهیم گرفت و یک نمونه ی خوب آن را روی مدل آموزش میدهیم و در نهایت تست میکنیم.

واژگان کلیدی: بورس - فارکس - هوش مصنوعی - بازار - پیش بینی - اقتصاد - بازار ارز - احتمال - شبکه عصبی - معامله

فهرست مطالب

صفحه	عنوان
۷	فصل اول: چرایی هوش مصنوعی
۷	۱-۱- چرایی هوش مصنوعی
۸	فصل دوم: بورس
۸	۱-۲- بورس
۸	۲-۲- بزرگترین بازارهای مالی
۸	۳-۲- فارکس
۹	فصل سوم: هوش مصنوعی
۹	۱-۳- هوش مصنوعی
۹	۲-۳- کاربردهای هوش مصنوعی در بورس
۹	۳-۳- چگونه هوش مصنوعی بورس را پیش‌بینی می‌کند؟
۱۰	۴-۳- شبکه عصبی مصنوعی
۱۱	فصل چهارم: فارکس و شبکه عصبی
۱۱	۱-۴- چرا فارکس؟
۱۱	۲-۴- نوع شبکه عصبی
۱۲	۳-۴- دیتا
۱۶	۴-۴- نویزگیری دیتا
۲۰	مراجع

فصل اول: چرایی هوش مصنوعی

۱-۱- چرایی هوش مصنوعی

شاید آگه بدونید وال استریت با استفاده از **هوش مصنوعی** و ابزارهای مهم آن می‌تواند میلیون‌ها داده‌ای را که در زمان واقعی اتفاق می‌افتد، جمع‌کند علاقتون برای خوندن این مطلب بیش از پیش شود. کارشناسان همیشه می‌گفتند به ثمر نشانیدن معاملات در وال استریت، بیشتر به هنر فرد بستگی دارد و موضوعی علمی نیست؛ حال این وضعیت در حال تغییر است. در ماه‌های اخیر، بورس و بازار سهام مورد توجه بسیاری از افراد در سراسر جهان قرار گرفته، البته وضعیت بازار در تمام کشورها مناسب نیست و مانند همیشه نوسانات شدیدی دارد، نوساناتی که شاید هوش مصنوعی در پیش بینی آنها موفق باشد.

به روایتی واژه بورس از نام خانوادگی شخصی به نام «واندر بورس» اخذ شده که در اوایل قرن پانزدهم میلادی در شهر بروژ، بلژیک می‌زیسته و صرافان شهر در مقابل خانه او گرد هم می‌آمدند و به دادوستد کالا، پول و اوراق بهادار می‌پرداختند. این نام بعدها به کلیه اماکنی اطلاق شد که محل دادوستد پول، کالا و اسناد مالی و تجاری بوده‌است [۳].

سال ۱۴۶۰ میلادی اولین مرکز بورس اوراق بهادار جهان، در شهر «انورس» بلژیک تأسیس شد. اما رسمیت بورس اوراق بهادار با انتشار سهام کمپانی هند شرقی هلند در سال ۱۶۰۲ میلادی در بورس آمستردام شکل گرفت. البته کمپانی هند شرقی هلند اولین شرکت سهامی به‌شمار نمی‌آید، بلکه اولین شرکت سهامی در سال ۱۵۵۳ میلادی با نام «مسکوی» در روسیه ایجاد شد [۳].

بدون شک، دنیای آینده، دنیای هوش مصنوعی و ربات‌ها خواهد بود. در این مقاله سعی کرده ایم که جای هوش مصنوعی در حوزه اقتصاد و بزرگترین آن یعنی بورس را نشان دهیم. از تجربیات بنده طی ۲ سال گذشته گرفته تا ترفندهای بزرگانی که شناخته شده‌ی این حوزه میباشند، در این مقاله بیان شده است. برای قابل فهم کردن این مقاله برای تمام دوست داران این حوزه نیز، ابتدا نگاهی مختصر به بورس، بزرگترین آن، هوش مصنوعی، به روزترین علم آن و درنهایت ادغام این دو علم بزرگ میپردازیم.

شروع و سرچشمه هوش مصنوعی به سال‌های جنگ جهانی دوم بر می‌گردد. زمانی که نیروهای آلمانی برای رمز نگاری و ارسال ایمن پیام‌ها از ماشین **enigma** استفاده می‌کردند و دانشمند انگلیسی، آلن تورینگ در تلاش برای شکست این کدها برآمد. تورینگ به همراه تیمش ماشین **bombe** را ساختند که **enigma** را رمز گشایی می‌کرد. هر دو ماشین **enigma** و **bombe** پایه‌های یادگیری ماشینی (**machine learning**) هستند که یکی از شاخه‌های هوش مصنوعی یا همان **Artificial intelligence** می‌باشد. تورینگ ماشینی را هوشمند می‌دانست که بدون اینکه به انسان حس صحبت با ماشین را بدهد، با او ارتباط برقرار کند و این مسئله پایه علم هوش مصنوعی است یعنی ساخت ماشینی که همانند انسان فکر، تصمیم‌گیری و عمل کند [۷].

رفته رفته با پیشرفت فناوری و سایر سخت افزارهای مورد نیاز برای توسعه هوش مصنوعی، ابزار هوشمند و سرویس‌های هوشمندی به بازار عرضه شدند که از هوش مصنوعی در بسیاری از فرآیندهایشان استفاده می‌کردند. بسیاری از سرویس‌های معروفی همانند موتورهای جستجو، ماهواره‌ها و ... از هوش مصنوعی استفاده می‌کردند. با معرفی گوشی‌های هوشمند و پس از آن گجت‌های هوشمند، هوش مصنوعی گام بلندی را برای ورود به زندگی انسان‌های پشت سر گذاشت. از این زمان به بعد هوش مصنوعی برای انسان‌ها جلوه کاربردی تری پیدا کرد و انسان‌ها بیشتر با واژه هوش مصنوعی و کاربردهای آن آشنا شدند.

در پایان یک مثال کاربردی برای متخصصان حوزه هوش مصنوعی و شبکه عصبی بیان میکنیم تا طرز کار ابتدایی را بیاموزند و ادامه دهنده این راه باشند. این نکته حائز اهمیت است که هدف ما در این مقاله بیان روشهای افزایش احتمالات در بورس است و لغت "پیش بینی" لغت مناسبی نیست چرا که بورس قابل پیش بینی نیست و درحال حاضر نیز بهترین شبکه‌های عصبی توانسته‌اند احتمال موفقیت را در این بازار به عدد ۶۰ درصد برسانند که اگر در زمینه بورس و معاملات آن تخصص داشته باشید، متوجه خواهید بود که با این احتمال ۶۰ درصد، میتوانید سرمایه بسیار بسیار زیادی بدست آورید و این عدد کم نیست. پس هدف ما نیز رسیدن به این عدد است و عدد صد که پیش بینی به حساب می‌آید را فراموش کنید.

فصل دوم: بورس

۲-۱. بورس

بازار بورس و سهام جایی است که در آن به خرید و فروش سهام و اوراق بهادار می‌پردازند. بالا و پایین رفتن ارزش این سهام‌ها، سود و زیان این معاملات را تعیین می‌کند. پر واضح است که بالا رفتن ارزش سهام و یا کاهش آن از یکسری قوانین و مشخصه‌هایی پیروی می‌کند که برای داشتن یک معامله‌ای پر سود، معامله‌گرها باید از استراتژی‌های پیچیده و تجزیه و تحلیل‌های زیادی استفاده کنند [۴].

۲-۲. بزرگترین بازارهای مالی

رتبه بندی زیر در سال ۲۰۱۹ انجام شده است و ۵ بازار بورس بزرگ و معتبر در این رتبه بندی آمده است. همانطور که قبلاً اشاره شد میزان بزرگی و یا اعتبار یک بورس به میزان سرمایه یا ارزش بازار آن بستگی دارد.

۱. بورس نیویورک New York Stock Exchange –
۲. بورس نزدک NASDAQ –
۳. بورس توکیو Tokyo Stock Exchange –
۴. بورس شانگهای Shanghai Stock Exchange –
۵. بازار بورس هنگ کنگ Hong Kong Stock Exchange –

همانطور که در رتبه بندی مشاهده می‌شود بازار بورس نیویورک (NYSE) بزرگ ترین بازار بورس در دنیا می‌باشد. اما علاوه بر این بازارها، بازار مالی دیگری نیز وجود دارد که اگر تمام معاملات روزانه این بازارها را هم روی یکدیگر بگذارید، به میزان معاملات روزانه آن بازار نمی‌رسند. بلکه بازار فارکس! [۵]

۲-۳. فارکس

بازار فارکس یک بازار جهانی نامتمرکز است که در آن همه ارزهای جهان در برابر یکدیگر معامله می‌شوند و معامله گران از تغییرات قیمت ارزها سود و زیان می‌کنند. بازار فارکس (Forex) تحت عناوین بازار مبادلات ارزهای خارجی یا FX یا بازار معامله ارز نیز شناخته می‌شود. بازار فارکس، بزرگترین بازار در جهان است که میانگین ارزش معاملات روزانه اش به بیش از ۵ تریلیون دلار در روز می‌رسد [۶].

فصل سوم: هوش مصنوعی

۳-۱. هوش مصنوعی

هوش مصنوعی که گاهی اوقات هوش ماشینی نامیده می‌شود، به هوشمندی نشان داده شده توسط ماشین‌ها در شرایط مختلف اطلاق می‌شود که در مقابل هوش طبیعی در انسان‌ها قرار دارد. به عبارت دیگر هوش مصنوعی به سامانه‌هایی گفته می‌شود که می‌توانند واکنش‌هایی مشابه رفتارهای هوشمند انسانی از جمله درک شرایط پیچیده، شبیه‌سازی فرایندهای تفکری و شیوه‌های استدلالی انسانی و پاسخ موفق به آنها، یادگیری و توانایی کسب دانش و استدلال برای حل مسایل را داشته باشند [۷].

در حوزه اقتصاد نیز همانطور که ذکر شد، در بازارهای بورس دنیا استفاده می‌شود و عملاً بدون هوش مصنوعی، با اینکه علم تازه و جدیدی است، دنیا در حوزه اقتصاد به این اندازه پیشرفت نمی‌کرد.

۳-۲. کاربردهای هوش مصنوعی در بورس

با توجه به تاثیراتی که گفته شد، استفاده از هوش مصنوعی می‌تواند منافع و کاربردهای زیادی برای معامله‌گران داشته باشد. تعدد سهم‌های موجود در بازار باعث می‌شود که معامله‌گران نتوانند آن‌ها را تحلیل کنند و استفاده از مزیت‌های هوش مصنوعی سبب می‌شود که بتوان حجم وسیع‌تری از سهام را تحلیل کرد و خرید و فروش نمود و این امر سبب داشتن معامله‌ای بهتر و سود بیشتر می‌شود.

به صورت کلی هوش مصنوعی می‌تواند در زمینه‌های زیر کاربرد داشته باشد [۹]:

- بدست آوردن استراتژی معاملاتی از طریق معاملات الگوریتمی
- نظارت بر بازار و تجزیه و تحلیل سریع آن
- پیش‌بینی روند پیش رو به کمک تجزیه و تحلیل داده‌ها گذشته
- افزایش الگو تریدینگ فرصت‌های معاملاتی برای داشتن خرید و فروش‌های کارآمد
- کمک به داشتن فرآیندی اتوماتیک و دقیق تر برای معاملات
- برخورداری از تغییرات زمان واقعی (real time) بازار
- مدیریت پرتفوی
- معامله الگوریتمی
- کشف کلاهبرداری مالی
- فرایند بیمه‌گری و وام‌دهی
- کشف الگوها
- معامله بر اساس اخبار اقتصادی
- سرعت بخشیدن به معامله‌گری

۳-۳. چگونه هوش مصنوعی بورس را پیش‌بینی می‌کند؟

جدیدترین فناوری‌هایی که کمک می‌کنند یادگیری ماشین و هوش مصنوعی بورس را محافظت کنند، شامل یادگیری عمیق (deep learning)، یادگیری انتقالی (transfer learning) و یادگیری انسان محور است. یادگیری عمیق به کامپیوترها اجازه می‌دهد که تعلیم ببینند، چه با نظارت و چه بدون نظارت انسان‌ها. این تعلیم دیدن با استفاده از الگوهای پیچیده، شبکه‌های عصبی (neural networks) و ارتباطات مخفی در مراکز داده‌ی عظیم شکل می‌گیرد. یادگیری انتقالی با نگاه به مدل‌سازی‌های شکل گرفته در گذشته، به خلق مدل‌های جدید می‌پردازد. از سوی دیگر یادگیری انسان محور اجازه می‌دهد که تحلیل‌گران بتوانند به طور موثر، تجربیات و تخصص خود را با ماشین‌ها در میان بگذارند. یادگیری انسان‌محور به متخصصان انسانی اجازه می‌دهد زمان و تلاش خود را صرف تحقیقات بازار و بررسی و راستی آزمایی شواهد کنند. امروزه

تیم نظارت بر بازار سهام نزدیک آمریکا سالانه بیش از ۷۵۰۰۰۰ هشدار را بررسی می‌کند و موارد مشکوک مانند حرکات غیر عادی قیمت‌ها، خطاهای معاملاتی و دستکاری‌های احتمالی را شناسایی می‌کند [۱۰].

شرکت‌هایی که با هوش مصنوعی بورس را تحلیل می‌کنند [۱۰]

• شرکت TRADING TECHNOLOGIES در شیکاگو

Trading Technologies صاحب یک پلتفرم هوش مصنوعی است که الگوهای معاملاتی پیچیده را در مقیاس عظیمی از بازارهای چندگانه‌ی مالی و بی‌درنگ (real-time) شناسایی می‌کند. با ترکیب فناوری یادگیری ماشین و قدرت پردازشی سریع داده‌های بزرگ (big data)، این شرکت به مشتریان خود ارزیابی مداومی از ریسک همکاری با آن‌ها را ارائه می‌دهد.

• شرکت GREENKEY TECHNOLOGIES در شیکاگو

این شرکت با استفاده از فناوری‌های تشخیص صدا (speech recognition) و پردازش زبان طبیعی (NLP) به کمک معامله‌گران می‌آید. این فناوری‌ها زمان زیادی برای معامله‌گران ذخیره می‌کنند و به جای آن‌ها در مکالمات، داده‌ها و صورت‌های مالی شرکت‌ها کاوش می‌کنند. با پلتفرم شرکت GreenKey Technologies، متخصصان مالی به کمک هوش مصنوعی می‌توانند بی‌درنگ (real-time) صورت‌های مالی شرکت‌ها را تحلیل کنند و شرکت‌های روبه‌رشد را شناسایی کنند.

• شرکت KAVOUT در سیاتل واشگتن

این شرکت محصولی به نام «K Score» دارد که خود زیرمجموعه‌ی پلتفرم هوش مصنوعی کای (kai) از همین شرکت است. K Score با پردازش مجموعه‌های عظیم و متنوع داده و اجرای مدل‌های پیش‌بینی کننده‌ی مختلف می‌تواند سهام شرکت‌ها را امتیازدهی کند. با استفاده از هوش مصنوعی، فناوری تشخیص الگو و موتور پیش‌بینی قیمت، Kavout هر روز بهترین شرکت‌های پیشنهادی را مشخص می‌کند.

• شرکت AI TRADING در لندن انگلستان

این استارت‌آپ با ترکیب هوش مصنوعی و جامعه‌ی معامله‌گری کمک می‌کند سود بیشتری کسب شود. AI Trading بازارهای مالی را رصد می‌کند تا بتواند فرصت‌های بهینه‌ی معامله‌گری را شکار کند. این معاملات با استفاده از قراردادهای هوشمند مبتنی بر بلاکچین انجام می‌شوند. همه‌ی معاملات وارد بلاکچین شده و غیر قابل تغییر و دستکاری هستند.

• شرکت SENTIENT INVESTMENT MANAGEMENT در سان‌فرانسیسکو کالیفرنیا

شرکت Sentient با استفاده از هوش مصنوعی هم در معامله‌گری و هم در سرمایه‌گذاری به طراحی استراتژی کمک می‌کند. با ترکیب فناوری‌های انقلابی و الگوریتم‌های یادگیری عمیق، این سیستم هوش مصنوعی به صورت مداوم پردازش می‌کند و استراتژی‌های جدیدی را توسعه می‌دهد.

۳-۴. شبکه عصبی مصنوعی

شبکه‌های عصبی مصنوعی یا به زبان ساده‌تر شبکه‌های عصبی سیستم‌ها و روش‌های محاسباتی نوین برای یادگیری ماشینی، نمایش دانش و در انتها اعمال دانش به دست آمده در جهت پیش‌بینی پاسخ‌های خروجی از سامانه‌های پیچیده هستند. ایده اصلی این گونه شبکه‌ها تا حدودی الهام‌گرفته از شیوه کارکرد سیستم عصبی زیستی برای پردازش داده‌ها و اطلاعات به منظور یادگیری و ایجاد دانش می‌باشد. عنصر کلیدی این ایده، ایجاد ساختارهایی جدید برای سامانه پردازش اطلاعات است [۸].

فصل چهارم: فارکس و شبکه عصبی

۴-۱. چرا فارکس؟

همانطور که میدانیم، شبکه عصبی الگوهای جامعه آماری خود را پیدا میکند، پس اگر جامعه ما الگویی نداشته باشد، شبکه عصبی کارآمد نخواهد بود. جامعه هایی دارای الگو میباشند که تعداد اعضای آن زیاد باشند، به عبارتی هیچ کس نتواند به راحتی در جامعه اثر گذار باشد و جهت حرکت جامعه را تغییر دهد. فارکس با حجم معاملات فوق العاده بالای خود، یک نمونه از جامعه ی بزرگ جهانی است که هیچ کس، اعم از بانک ها موسسات مالی بزرگ، نمیتوانند جهت آن را تغییر دهند. پس بهترین بازار برای افزایش احتمالات موفقیت با استفاده از شبکه عصبی، بازار فارکس خواهد بود. مسئله ای که در ابتدای مقاله، اشاره ای شد. [۱] اما در فارکس نیز ما جفت ارزهای مختلفی داریم که برای نتیجه بهتر و مطمئن تر، پرحجم ترین آن یعنی eurUSD را انتخاب میکنیم.

۴-۲. نوع شبکه عصبی:

معامله گران بزرگ میدانند که نتیجه موفق هر معامله، حاصل دو عمل میباشد: ورود خوب و خروج بهتر! به عبارتی، تنها انتخاب معامله خوب، نتیجه ی عالی در پی نخواهد داشت و خروج نیز بسیار مهم است. ما برای ورود، هدف یا تارگت داریم، به عبارتی هدف ما از قبل مشخص است و میتوانیم خود ما نیز تعیین کنیم که مثلاً در فلان زمان، بهترین معامله کدام است، خرید یا فروش. پس ما برای آموزش شبکه عصبی برای ورود به معامله، از شبکه عصبی های هدفدار استفاده خواهیم کرد و از آنجا که دیتای فارکس یا بورس، دارای سری زمانی میباشد، پس بهترین لایه برای آن نیز، LSTM یا gru خواهد بود. اما برای خروج از معامله، ما از قبل هدفی نداریم، یا به عبارتی خود تریدر نیز نمیداند که بهترین خروج، چه زمانی میباشد، پس برای موفقیت در این زمینه، از شبکه عصبی تقویتی استفاده میکنیم که برای لایه های آن نیز، از همان لایه های LSTM استفاده میکنیم. از آنجا که هردوی اینها، نیازمند یک مقاله جداگونه میباشد و ما سعی داریم به یک گوشه ای از طرز کار با آنها آشنا شویم و این مقاله گنجایش حجم هر دو شبکه را ندارد، یکی از انواع این شبکه ها را بررسی میکنیم. از آنجا که طبق گفته بزرگان بازار، حدود ۸۰ درصد موفقیت در یک معامله به ورود بستگی دارد، پس ما نیز شبکه عصبی هدفدار برای بخش ورود معامله یعنی خرید یا فروش را بررسی میکنیم.

سخت افزار:

برای نتیجه بهتر و یکدستگی، از سخت افزار رایگان گوگل استفاده میکنیم. [Colab.research.google.com](https://colab.research.google.com) این فضای ابری، به شما امکان دسترسی و استفاده از چهار نوع gpu مختلف برای کارهای تحقیقاتی خود را میدهد.

```
[{"name": "/device:CPU:0",
  device_type: "CPU",
  memory_limit: 268435456,
  locality {
  },
  incarnation: 826175317514798981,
  name: "/device:XLA_CPU:0",
  device_type: "XLA_CPU",
  memory_limit: 17179869184,
  locality {
  },
  incarnation: 12695409380329768766,
  physical_device_desc: "device: XLA_CPU device",
  name: "/device:XLA_GPU:0",
  device_type: "XLA_GPU",
  memory_limit: 17179869184,
  locality {
  },
  incarnation: 9078804620300355446,
  physical_device_desc: "device: XLA_GPU device",
  name: "/device:GPU:0",
  device_type: "GPU",
  memory_limit: 15701340352,
  locality {
    bus_id: 1
    links {
    }
  },
  incarnation: 13899808492076694945,
  physical_device_desc: "device: 0, name: Tesla P100-PCIE-16GB, pci bus id: 0000:00:04:0, compute capability: 6.0"
}]
```

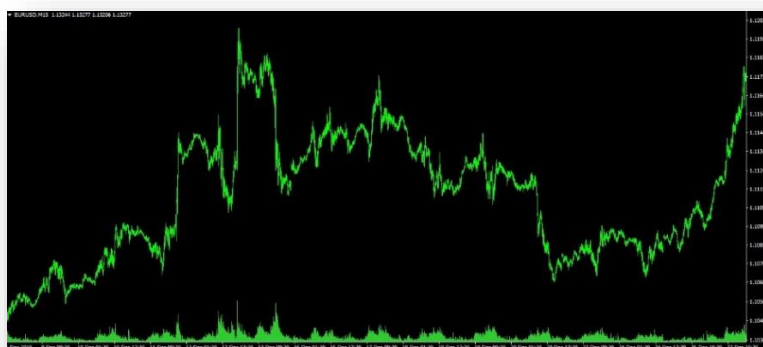
۳-۴. دیتا

بدون شک، مهمترین مسئله، حتی مهمتر از شبکه عصبی و هوش مصنوعی، مسئله ی دیتاست. پیدا کردن دیتای مناسب، مهمترین مشکل تمام فعالان این حوزه است. حتی در زمینه های مختلف هوش مصنوعی، دیتا یکی از اصلی ترین مسائل تمام اندیشمندان است.

پس در این مقاله نیز، توجه ویژه ای به مسئله ی دیتای ورودی شبکه عصبی خواهیم داشت و من در زیر، به بررسی تمام تجربیات خود در این زمینه میپردازم. در انتها نیز یکی از بهترین نوع دیتا را به شبکه عصبی می‌دهیم و آموزش مدل و پیش بینی را مرور میکنیم. لازم به ذکر است که ما دیتای خود را از معروفترین نرم افزار فعالان فارکس، یعنی متاتریدر خواهیم گرفت.

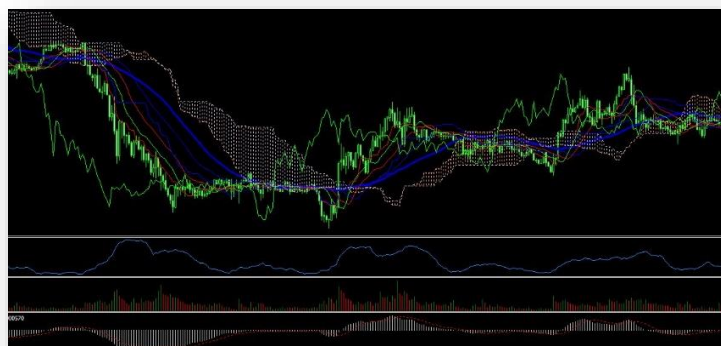
حالت اول:

اولین دیتایی که اکثر شما به ذهنتان میرسد، خود قیمت است. بله میتوانیم خود قیمت جفت ارز را به شبکه عصبی بدهیم و قیمت یک کندل بعد را به عنوان تارگت، از شبکه بخواهیم. حتی میتوانیم حجم را نیز به آن اضافه کنیم.



اما خود قیمت نمیتواند نتیجه مطلوبی را به ما بدهد. یکی از دلایل آن بدون شک، نویز بسیار زیاد این نوع دیتاست. دلیل دیگر نیز عدم وجود جزئیاتی همچون سطوح حمایت مقاومت است.

حالت دوم: دومین نوع دیتایی که به ذهن تمام شما میرسد، استفاده از اندیکاتور هاست. بله اندیکاتورها میتوانند ایده خوبی باشند ولی کافی نیستند. یکی از دلایل ضعف آنها، فاصله زمانی یا به عبارتی delay است. دلیل بعدی میتواند عدم وابستگی بازار به آنها باشد که در این صورت شبکه ما ممکن است کمی گیج شود و به راحتی به هدف نرسد.



حالت سوم: یکی از ایده های ناب در این زمینه، استفاده از دیتا در زمان های مختلف است. به عبارتی یک دیتا در تایم فریم ۵ دقیقه ای، یکی در تایم ۱۵، دیگری در ۶۰ دقیقه ای. این ایده ی خوبی است و میتواند مشکل سطوح حمایت مقاومت را تا حدودی برطرف کند. میتوانیم از اندیکاتورهای مفید نیز استفاده کنیم

در زیر ایده های به کار رفته را به طور مختصر بیان میکنیم:
-دیتای تیک-tick همراه با حجم لحظه ای به عنوان ورودی: دیتای تیک میتواند زمان و میزان حجم ورودی بزرگان بازار را به خوبی نشان دهد. اما مشکل اصلی، دسترسی به دیتای تیک صحیح همراه با حجم واقعی است. هیچ بروکری این دیتا را در دسترس بقیه قرار نمیدهد. دیتاهای رایگان نرم افزار های مختلف نیز، کامل نیستند.

-استفاده از انواع اندیکاتورها(همانطور که در بالا ذکر شد)

-استفاده از open و close و high و low به عنوان الگوهای کندلی یا سطوح حمایت مقاومت(این سطوح در تایم بالا، نقش سطوح حمایت مقاومت در تایم پایین تر را ایفا میکنند.)

-استفاده از جفت ارزهای مختلف:میدانیم که ارز مهم eurUSD کمتر از ۳۰ درصد حجم معاملات را شامل میشود. پس برای پیش بینی این ارز، باید از ارزهای مختلف نیز استفاده کنیم. تمام ارزهای وابسته به euro یا dollar میتوانند در سرنوشت جفت ارز eurUSD نقش خود را ایفا کنند.

-استفاده از تایم زمانی مختلف: هر تریدر میداند که برای ترید، باید یک تایم تحلیل، یک تایم ورود و یک تایم خروج داشته باشد. پس مجبوریم از تایم های مختلف استفاده کنیم. شاید فکر کنید که کافیست دیتای تایم پایین را زیاد کنیم تا به دیتای تایم بالا برسیم، اما این کار نه تنها مفید نیست، بلکه باعث میشود شبکه ما وارد جزئیات شود و دچار overfit شویم.

-استفاده از عکس بازار: ایده جالبی است که از عکسها با پیکسل های مشابه و ارائه آن به شبکه عصبی کانولوشن برای پیش بینی بازار استفاده کنیم. اما این کار نیز مفید نخواهد بود. یکی از دلایل آن، نویزهای زمانی و قیمتی بازار است. منظور از نویز زمانی آن است که بازار ممکن است به دیتای ۱۰۰ کندل قبل خود وابسته باشد یا به دیتای ۱۰۰ کندل قبل خود. پس عکس با پیکسل های مشابه نیز کارآمد نخواهد بود.

-استفاده از الگوهای کندلی

-استفاده از اندیکاتورهای شخصی

نتیجه گیری دیتا: در کل بهترین دیتای ما، باید ویژگی های زیر را دارا باشد:

مولتی تایم: تایم فریم های مختلف را شامل شود.

مولتی ارز: ارزهای مختلفی که شامل دلار یا یورو هستند برای پیش بینی eurUSD لازم میباشد.

بدون نویز: تا حد ممکن به صورت دستی نویز گیری شود و در انتها به وسیله ی شبکه ی اتوانکدر دوباره نویزگیری شوند.

سطوح حمایت مقاومت: بهترین دیتا به در نظر گرفتن سطوح حمایت مقاومت، حتی ۵ درصد هم ما را به هدفمان نمیرساند. پس این نکته مهم را در نظر بگیرید و سعی کنید بهترین سطوح را به شبکه بدهید.

حجم: حجم بسیار مفید است و به طور نامحسوسی نویز گیری نیز میکند.

یک نمونه از دیتا:

Eurusd.M۵.MovingAverage(۶)*۶
Usdjpy.M۵.MovingAverage(۶)*۶

Eurusd.M۱۵.MovingAverage(۶)*۶
Usdjpy.M۱۵.MovingAverage(۶)*۶

Gbpusd.M^o.MovingAverage(٦)*٦
 Audusd.M^o.MovingAverage(٦)*٦
 Usdcad.M^o.MovingAverage(٦)*٦
 Usdchf.M^o.MovingAverage(٦)*٦
 Nzdsd.M^o.MovingAverage(٦)*٦
 Eurjpy.M^o.MovingAverage(٦)*٦
 Eurgbp.M^o.MovingAverage(٦)*٦
 Euraud.M^o.MovingAverage(٦)*٦
 Eurcad.M^o.MovingAverage(٦)*٦
 Eurchf.M^o.MovingAverage(٦)*٦
 Eurnzd.M^o.MovingAverage(٦)*٦

Gbpusd.M^١.MovingAverage(٦)*٦
 Audusd.M^١.MovingAverage(٦)*٦
 Usdcad.M^١.MovingAverage(٦)*٦
 Usdchf.M^١.MovingAverage(٦)*٦
 Nzdsd.M^١.MovingAverage(٦)*٦
 Eurjpy.M^١.MovingAverage(٦)*٦
 Eurgbp.M^١.MovingAverage(٦)*٦
 Euraud.M^١.MovingAverage(٦)*٦
 Eurcad.M^١.MovingAverage(٦)*٦
 Eurchf.M^١.MovingAverage(٦)*٦
 Eurnzd.M^١.MovingAverage(٦)*٦

Eurusd.H^١.MovingAverage(٦)*٦
 Usdjpy.H^١.MovingAverage(٦)*٦
 Gbpusd.H^١.MovingAverage(٦)*٦
 Audusd.H^١.MovingAverage(٦)*٦
 Usdcad.H^١.MovingAverage(٦)*٦
 Usdchf.H^١.MovingAverage(٦)*٦
 Nzdsd.H^١.MovingAverage(٦)*٦
 Eurjpy.H^١.MovingAverage(٦)*٦
 Eurgbp.H^١.MovingAverage(٦)*٦
 Euraud.H^١.MovingAverage(٦)*٦
 Eurcad.H^١.MovingAverage(٦)*٦
 Eurchf.H^١.MovingAverage(٦)*٦
 Eurnzd.H^١.MovingAverage(٦)*٦

Eurusd.H^٤.MovingAverage(٦)*٦
 Usdjpy.H^٤.MovingAverage(٦)*٦
 Gbpusd.H^٤.MovingAverage(٦)*٦
 Audusd.H^٤.MovingAverage(٦)*٦
 Usdcad.H^٤.MovingAverage(٦)*٦
 Usdchf.H^٤.MovingAverage(٦)*٦
 Nzdsd.H^٤.MovingAverage(٦)*٦
 Eurjpy.H^٤.MovingAverage(٦)*٦
 Eurgbp.H^٤.MovingAverage(٦)*٦
 Euraud.H^٤.MovingAverage(٦)*٦
 Eurcad.H^٤.MovingAverage(٦)*٦
 Eurchf.H^٤.MovingAverage(٦)*٦
 Eurnzd.H^٤.MovingAverage(٦)*٦

Eurusd.D^١.High[٠]*٦ Eurusd.D^١.Low[٠]*٦ Eurusd.Close.High[٠]*٦

Eurusd.Volume*٦

چند نکته:

- حجم ما با کاملتر و واقعی تر باشد. حجم بروکر های را با یکدیگر جمع کنید تا به حجم واقعی تر دست پیدا کنید.
- سطوح حمایت مقاومت بستگی به میزان تارگت شما دارد. این یک نمونه دیتا برای پیش بینی کوتاه مدت است.
- ۱۳ ارز برای پیش بینی ارز ما کافی است اما میتوانید برای نتیجه بهتر از طلا و شاخص دلار یا نفت نیز استفاده کنید.

جمع بندی دیتا: دیتای ما شامل ۵۶ ستون ۶ تایی است. هرچه تعداد نمونه های ما بیشتر باشد، میتوانیم انتظار نتیجه بهتر را نیز داشته باشیم. پس دیتای تایم ۵ دقیقه ای ارز یورو دلار را از سال ۲۰۱۰ تا ۲۰۲۰ را از نرم افزار متاتریدر خارج میکنیم که تقریباً شامل ۸۰۰۰۰۰ نمونه است.

۸۰۰۰۰۰*۵۶*۶*

حال دیتای ورودی ما آماده است.

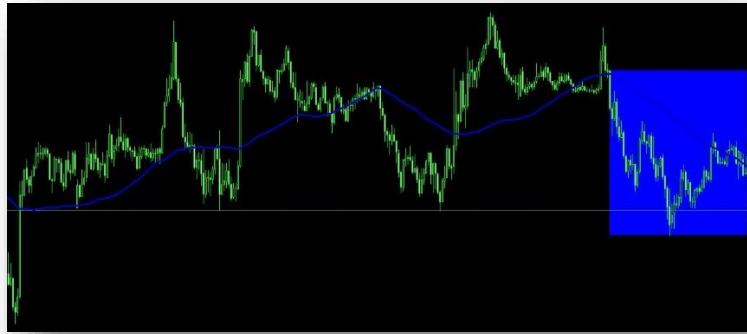
تارگت:

از خود دیتای ورودی مهمتر، دیتای خروجی یا تارگت ماست. تارگت باید خیلی خوشمندانانه و متناسب با دیتای ورودی ما باشد.

از آنجا که جنس ورودی ما مووینگ اوریج میباشد، پس خروجی ما نیز باید دوره ای از مووینگ اوریج باشد، اما مووینگ با پریود کم، نمیتواند مورد استفاده قرار بگیرد، چرا که ورودی ما خود دارای delay با دوره ۶ میباشد، پس تارگت باید دوره ای داشته باشد که عدد، در آن گم شود. حداقل دوره، برابر ۶۰ میباشد. یعنی پیش بینی ۵ ساعت آینده یا ۶۰ کندل پنج دقیقه ای دیگر.

Target: EurUSD.M^۵.MovingAverage(۶۰)

دیتای تارگت ما نیز ۸۰۰۰۰۰ واحد میباشد.



چند نکته:

- ۱- تنظیمات MovingAverage در بهترین حالت عبارت است از:
MA.metod.Exponential
Apply to Typical Price(HLC/۳)
- ۲- دیتای قبل سال ۲۰۱۰ چندان تکنیکال نبوده (به دلیل حجم کم معاملات)، پس سعی کنید از دیتای سال ۲۰۱۰ به بعد استفاده کنید.
- ۳- میتوانید دیتای مووینگ اوریج روزانه را نیز به دیتای ورودی خود اضافه کنید.
- ۴- هرچه حجم واقعی تر باشد، نتیجه بهتری کسب خواهیم کرد.
- ۵- معتبرترین و بهترین خطوط حمایت مقاومت براساس قیمت high,low,close ، برای تایم روزانه میباشد. پس تایم فریم این دیتای ورودی را تغییر ندهید.

آموزش دیتا:

حال که دیتای ما کامل است، برای آموزش آماده میباشیم و در این مرحله ما ۳ بخش کلی خواهیم داشت: پیش برآزش، برآزش و تست. پیش برآزش: این مرحله در حقیقت آماده کردن دیتا برای برآزش یا آموزش میباشد. یک سری عملیات ها باید روی دیتا صورت بگیرد تا قابل آموزش باشد. ابتدا باید دیتا را نرمالایز کنیم. به عبارتی دیتاها را هماهنگ و یکدست کنیم تا شبکه بهتر و راحتتر آموزش ببیند. نرمالیزه کردن دیتای نوع ۱ یعنی مووینگ اوریج: جنس این دیتا، قیمت است پس میتوان به روش زیر این دیتا را نرمالیزه کرد:

```
for i in range(datax.shape[0]):
    for j in range(0,52):
        ma=max(datax[i,:,j])
        mi=min(datax[i,:,j])
        datax[i,:,j]=(datax[i,:,j]-mi)/(ma-mi)
```

-تعداد دیتای از نوع قیمت ما، ۵۲ در ۶ میباشد.

-زبان برنامه نویسی به کار رفته در این پروژه، پایتون میباشد.

نرمالیزه کردن دیتای حمایت مقاومت: این دیتا را ابتدا باید به پیپ یا بیت (پیس) از متاتریدر دریاوریم. یعنی اختلاف این قیمت با قیمت فعلی را از متاتریدر بیرون بکشیم. پس دیتای ما یک عدد خواهد بود. مثبت و منفی برای نرمالیزه کردن این دیتا، کافیهست که بازه کل دیتا را داشته باشیم، بعد کل دیتا را تقسیم بر آن عدد بکنیم تا بازه این دیتا بین منفی یک و یک بشود. به عبارتی اگر این دیتای ما در کل بین منفی هزار تا هزار است، کل دیتا را بر عدد هزار تقسیم میکنیم.

نرمالیزه کردن دیتای حجم: این دیتا یک عدد مثبت خواهد بود. پس بهترین روش برای آن، روش زیر است:

```
for i in range(datax.shape[0]):
    for h in range(input_len-1,-1,-1):
        datax[i,h,vol]=(datax[i,h,vol]/datax[i,0,vol])-1
```

مرحله ی بعد از نرمالایز کردن، تقسیم دیتا به دودسته آموزش و تست میباشد. بهتر است که دیتای آموزش ۸۰ درصد باشد.

```
train_len=datax.shape[0]*0.8
datax_train=datax[:train_len,:,:)
datay_train=datay[:train_len]
datax_test =datax[train_len:,:,:)
datay_test =datay[train_len:]
```

۴-۴. نويز گيری دیتا:

در این قسمت ما به ۲ نوع میتوانیم دیتای خود را نويز گيری کنیم که هر روش، تخصص خاص خود را دارد. در روش اول، میتوانیم دیتا را به طور مستقیم نويز گيری کنیم که نیاز به تخصص در فارکس میباشد. در زیر چند روش را بیان میکنیم.

۱- خود دیتا را بررسی، و اعداد پرت را از بین میبریم یا نرمالیز میکنیم. برای مثال اگر دیتای نوع دوم ما که پیپ بود، میتوانیم دیتاهای پرت را به اعداد پایین تر کاهش دهیم تا با این کار، بازه اعداد نیز کاهش یابد.

۲- خود بازار فارکس، در برخی تایم ها، به دلیل حجم کم، دارای نويز زیادی میباشد. میتوانیم این تایم ها را به طور کلی حذف کنیم. چرا که به دلیل حجم پایین، نويز محسوب میشوند. این تایم ها، زمان بسته بودن بازار اتحادیه اروپا و آمریکا است.

۳- تنظیمات مووینگ اوریج، خود به کاهش نويز کمک میکند. آن تنظیماتی که در بالا گفته شد، برای کاهش نويز عالی میباشد.

۴- تغییرات زیاد بازار ناشی از اخبار، یک نويز میباشد. چرا که ما با تحلیل تکنیکال قصد پیش بینی بازار را داریم. پس اخبار که در حوزه تحلیل فاندamental میباشد، برای ما نويز محسوب میشود. اگر بتوانید این دیتا را حذف کنید، به کاهش نويز، کمک میکنید.

کاهش نویز با شبکه عصبی: بهترین شبکه برای این منظور، اتوانکر میباشد که با این روش میتوانیم تعداد دیتا را نیز کاهش دهیم. این روش خود یک شبکه عصبی است که نیاز به یک مقاله جداگانه دارد و در این مقاله نمیگنجد.

پرازش: در این مرحله ما به آموزش دیتا میپردازیم. آخرین مرحله مشکل و حساس، این مرحله است. قبل از بررسی این روش، به بررسی سیستم سخت افزاری خود میپردازیم.

سخت افزار به کار رفته: برای تحلیل سریع در شبکه عصبی و به طور کلی، معمولاً از gpu استفاده میشود. (گاهی از tpu نیز بهره میگیریم).

همان طور که ذکر شد، سیستم ما، سیستم رایگانی است که گوگل در اختیار اندیشمندان حوزه هوش مصنوعی قرار داده است. به نشانی:

Colab.research.google.com

توجه کنید که این سایت، ۴ gpu مختلف در اختیار کاربران خود قرار میدهد که برای نتیجه بهتر، روی بهترین آن یعنی ۴t تنظیم کنید. برای حل مشکل رم به دلیل دیتای نمونه ی زیاد، رم ۲۰ گیگ مفید را انتخاب کنید.

پرازش: همان طور که قبلاً هم ذکر شد، بهترین شبکه و درحقیقت بهترین لایه، لایه های lstm میباشد. چرا که دیتای ما، به صورت سری زمانی میباشد و در هر لحظه آپدیت میشود. بهترین شبکه ای که بنابر تجربه بدست آمده است، شبکه زیر میباشد:

```
inputlayer= tf.keras.layers.Input (shape=(input_len,in2_len
layer      = tf.keras.layers.LSTM (neuron ,return_sequences=True,recurrent_activation='sigmoid')(inputlayer)
layer      = tf.keras.layers.Dropout(Dropout
layer      = tf.keras.layers.LSTM (neuron ,return_sequences=True,recurrent_activation='sigmoid')( layer)
layer      = tf.keras.layers.LSTM (neuron ,recurrent_activation='sigmoid')( layer)
layer      = tf.keras.layers.Dropout(Dropout
layer      = tf.keras.layers.Dense (1 ,activation="relu"
model      = tf.keras.Model (inputlayer,layer
```

چند نکته:

برای جلوگیری از overfit ، از لایه های dropout بهره میگیریم، تنظیمات اولیه این لایه را روی عدد ۰/۲۵ تنظیم میکنیم. میتوانیم بنابر تجربه بعد چندین اپیوچ آموزش، تنظیمات آن را تغییر دهیم.

تابع فعالسازی نیز در لایه های lstm در بهترین حالت و بنابر تجربه، sigmoid بدست آمده است. در لایه آخر نیز، بهتر است که روی relu تنظیم شود. چرا که خروجی ما خود یک عدد مثبت خواهد بود. توجه کنید که پایه ی شبکه عصبی، سعی و خطا و کاهش خطاست، برای دستیابی به بهترین تنظیمات جزئیات شبکه نیز، باید از روش سعی و خطا استفاده کرد.

```
model.compile(optimizer=tf.keras.optimizers.Adam(),loss=tf.keras.losses.mse)
```

تابع خطا: بهترین حالت برای نتیجه، نتایج با فاصله کم خواهد بود، پس بهتر است که از تابع mse بهره بگیریم. چرا که این تابع سعی میکند خطاهای بزرگ را کاهش دهد.

اپتیمایزر: بهترین اپتیمایزر نیز بنابر تجربه، adam خواهد بود.

آموزش: حال با تابع fit، مدل را آموزش میدهیم و نتیجه را مشاهده میکنیم:

```

for i in range(40):
    print(i)
    history=model.fit(datax_train,datay_train,batch_size=128,validation_data=(datax_test,datay_test),epochs=1)
    q='cat /content/gdrive/My Drive/Save_models/lstm2.txt'
    if (float(q[0]))>(history.history['val_loss'][0]):
        qq=str(history.history['val_loss'])
        with open('/content/gdrive/My Drive/Save_models/lstm2.txt', 'w') as f:
            f.write(str(history.history['val_loss'][0]))
        model.save('/content/gdrive/My Drive/Save_models/multiLstm2.h5')

```

نکته: برای اینکه بهترین حالت را ذخیره کنیم، بعد از بدست آوردن کمترین خطای تست، آن را ذخیره میکنیم. بچ سائز نیز در بهترین حالت، ۱۲۸ بدست آمده است.

حال نتیجه آموزش را براساس خطایی که کراس محاسبه میکند، نشان میدهیم:

```

0
782/782 [=====] - 32s 41ms/step - loss: 65230.1250 - val_loss: 26586.8105
1
782/782 [=====] - 31s 39ms/step - loss: 62054.7852 - val_loss: 24709.3262
2
782/782 [=====] - 31s 39ms/step - loss: 59276.0781 - val_loss: 23162.0723
3
782/782 [=====] - 31s 39ms/step - loss: 56803.4336 - val_loss: 21786.0605
4
782/782 [=====] - 31s 39ms/step - loss: 54488.5000 - val_loss: 20555.8047
5
782/782 [=====] - 30s 39ms/step - loss: 52357.0234 - val_loss: 19462.0234
6
782/782 [=====] - 30s 39ms/step - loss: 50420.8164 - val_loss: 18499.5078
7
782/782 [=====] - 30s 39ms/step - loss: 48650.5742 - val_loss: 17668.3086
8
782/782 [=====] - 30s 39ms/step - loss: 46997.6016 - val_loss: 16958.4512
9
782/782 [=====] - 30s 39ms/step - loss: 45494.5781 - val_loss: 16368.1152
10
782/782 [=====] - 30s 39ms/step - loss: 44028.1250 - val_loss: 15893.9756
11
782/782 [=====] - 30s 39ms/step - loss: 42789.7812 - val_loss: 15535.5938
12
782/782 [=====] - 30s 39ms/step - loss: 41717.9297 - val_loss: 15285.4980
13
782/782 [=====] - 30s 39ms/step - loss: 40770.0547 - val_loss: 15138.8506
14
782/782 [=====] - 30s 39ms/step - loss: 40037.1094 - val_loss: 15088.2451
15
782/782 [=====] - 30s 39ms/step - loss: 39338.2539 - val_loss: 15124.4912
16
782/782 [=====] - 30s 39ms/step - loss: 38786.2148 - val_loss: 15235.5127

```

بهترین خطای تست، عدد ۱۵۰۰۰ بدست آمده است که باید این خطا را در عمل تست کنیم. ابتدا این خطا را در خود پایتون و با تابع اختصاصی بررسی میکنیم، کافیت تابعی بسازیم که بالا یا پایین بودن تارگت را بررسی کند. به عبارتی بررسی کنیم که buy یا sell را به درستی نشان میدهد یا نه.

```

def testpredict(prte):
    t,f,su=0,0,0
    for i in range(datay_test.shape[0]):
        if (prte[i])>=0 and (datay_test[i])>=0:
            su=su+datay_test[i,0]
        elif (prte[i]<0 and (datay_test[i])<0:
            t=t+1
        else:
            f=f+1
    if t>f:
        print(testpredict(model.predict(datax_test)))

```

در پایان مشاهده خواهیم کرد که این تابع، عدد ۵۴ درصد را نشان میدهد. به عبارتی فقط ۵۴ درصد مواقع، میتواند درست پیش بینی کند. شاید بیان کنید که ۵۴ با ۶۰ درصد بهترین شبکه فاصله چندانی ندارد، ولی باید بدانید که این درصدها، در عملی تصاعدی عمل میکنند. به عبارتی اختلاف نتیجه عملکرد ۵۴ با ۵۵ درصد در عمل، خیلی کمتر از اختلاف نتیجه عملکرد ۵۷ با ۵۸ درصد خواهد بود. پس این شبکه فاصله زیادی با شبکه برتر یعنی ۶۰ درصد خواهد داشت.

تست در عمل

برای اینکه نتیجه کار خود را بررسی کنیم و ببینیم در هر زمان، شبکه عصبی ما چه تصمیمی میگیرد، لازم است که همین مدل شبکه عصبی خود را در نرم افزار متاتریدر پیاده کنیم. این کار نیازمند دانش برنامه نویسی به زبان `mql` یعنی زبان برنامه نویسی مخصوص نرم افزار متاتریدر میباشد که بحث آن در این مقاله نمیگنجد. پس از ساخت شبکه، دیتایی که به شبکه هنگام آموزش مدل دادیم، همان دیتا را منتها در زمان حال و جدید، به شبکه میدهیم تا عمل پیش بینی را انجام دهد سپس نتیجه کار را به صورت آنلاین بررسی میکنیم. یکی از بهترین قسمت های نرم افزار متاتریدر برای این عمل، قسمت `tester` آن است که مختص ربات ها ساخته شده است:



نتیجه گیری

دیدیم که با این ورودی و خروجی و شبکه، توانستیم تا ۵۴ درصد، حرکت صعودی یا نزولی بازار در ۵ تا ۶ ساعت آینده را به درستی پیش بینی کنیم. برای حصول نتیجه بهتر، باید بیشتر روی دیتای ورودی خود کار کنید سپس روی دیتای تارگت و در نهایت با بهبود شبکه عصبی خود، بهترین نتیجه ممکن را بدست آورید. این نکته را در نظر داشته باشید که برای بدست آوردن بهترین نتیجه، باید شبکه های مختلف زیادی را تست کنید تا در هر شبکه، تجربه خوبی کسب کنید و نتیجه را بهبود ببخشید.



Faculty of Engineering
Department of Computer

B.Sc. Final Project Report

Title of the Report:

Increase the chances in the stock market by using artificial intelligence

Under Supervision of:

Professor Razavi Ebrahimi

By:

Mohammad ahmadian honarmand

DATE:

Winter-۲۰۲۱

