بسم الله الرحمن الرحيم

دانشگاه صنعتی شریف
دانشکده ریاضی

پروژه درس ریاضی مالی

آپشن‌های قیمت‌گذاری و محاسبه نوسانات ضمنی با استفاده از شبکه‌های عصبی

نگارش

نازنین میرزائی

سیدرضا توکلی

استاد درس

دکتر فتوحی

تیر ۱۴۰۲

چکیده

در این گزارش کار، محتوا و کاربرد مقاله " **Pricing Options and Computing Implied Volatilities using Neural Networks**" به وسیله اجرا و بررسی کدهای شبیه سازی مرتبط ارزیابی شده است.
این مقاله روش استفاده از شبکه های عصبی برای تخمین شاخص نوسان ضمنی و قیمت گذاری آپشن با مدل های بلک شولز و هستون را توصیف می کند.
اهداف این گزارش عبارتند از:
۱) بررسی روش ها و الگوریتم های ارائه شده در مقاله
۲)اجرا و شبیه سازی کدها برای بررسی عملکرد الگوریتم ها
۳) ارائه نتایج و نکات اصلی مقاله

کلیدواژه‌ها: نوسان ضمنی، مدل بلک شولز، مدل هستون

فهرست مطالب

مقدمه

آپشن های قیمت گذاری ابزارهایی مهم برای مدیریت ریسک و کنترل قیمت‌های دارایی‌های مالی هستند. به کمک این آپشن ها، سرمایه گذاران و شرکت ها می‌توانند از افزایش قیمت‌ها و کاهش آن‌ها محافظت کرده و ریسک خود را کنترل کنند.

تاریخچه قیمت گذاری اختیار معامله و نوسانات ضمنی به اوایل قرن بیستم باز می گردد. در سال ۱۹۰۰، لویی باکلیه، ریاضیدان فرانسوی، پایان نامه دکترای خود را با عنوان "نظریه سفته بازی" منتشر کرد که در آن مفهوم فرآیندهای تصادفی برای مدل سازی قیمت سهام معرفی شد. با این حال، کار او تا حد زیادی برای سال‌ها نادیده گرفته شد و تنها در دهه‌های ۱۹۵۰ و ۱۹۶۰ بود که ایده‌های او به رسمیت شناخته شد. در دهه ۱۹۷۰، مدل بلک اسکولز توسط فیشر بلک، مایرون اسکولز و رابرت مرتون توسعه یافت. این مدل فرمولی برای قیمت‌گذاری آپشن‌های به سبک اروپایی ارائه می‌دهد و بر اساس فرضیات نوسانات ثابت و بازارهای کارآمد است. مدل Black-Scholes بازار آپشن ها را متحول کرد و امکان قیمت گذاری دقیق و سریع آپشن ها را فراهم کرد. با این حال، مدل بلک شولز این واقعیت را در نظر نگرفت که قیمت سهام همیشه ثابت نیست و نوسان می‌تواند در طول زمان تغییر کند. در دهه ۱۹۸۰، مفهوم نوسانات ضمنی برای توضیح این موضوع معرفی شد. نوسانات ضمنی نوسانی است که توسط قیمت فعلی یک آپشن در بازار وجود دارد. معامله گران از نوسانات ضمنی برای ارزیابی انتظارات بازار از نوسانات آتی استفاده می کنند و استراتژی های معاملاتی خود را بر این اساس تنظیم می کنند. در دهه ۱۹۹۰، مدل بلک-اسکولز برای گنجاندن مفهوم نوسانات تصادفی توسعه یافت که امکان مدل‌سازی واقعی‌تر قیمت سهام را فراهم می‌کرد. معرفی مدل های نوسانات تصادفی، مانند مدل هستون، امکان قیمت گذاری دقیق تر آپشن‌ها و مدیریت ریسک بهتر را برای معامله گران فراهم کرد.

قیمت گذاری اختیار معامله فرآیند تعیین ارزش منصفانه یک آپشن است. اختیار معامله یک قرارداد مالی است که به مالک این حق را می دهد، اما نه تعهدی، برای خرید یا فروش یک دارایی پایه با قیمتی مشخص، که به عنوان قیمت اعتصاب شناخته می شود، در یک دوره زمانی مشخص. آپشن‌ها معمولاً برای پوشش ریسک، سفته بازی و تولید درآمد استفاده می شوند. قیمت یک آپشن تحت تأثیر عوامل مختلفی است، از جمله:

۱. قیمت فعلی دارایی پایه: قیمت یک اختیار معامله مستقیماً با قیمت دارایی پایه مرتبط است. با افزایش قیمت دارایی پایه، ارزش اختیار خرید (که به مالک حق خرید دارایی را می دهد) افزایش می یابد و ارزش اختیار فروش (که به مالک حق فروش دارایی را می دهد) کاهش می

یابد.

۲. قیمت اعتصاب: قیمت اعتصاب قیمتی است که دارایی پایه را می توان با آن خرید یا فروخت. ارزش یک اختیار خرید با افزایش قیمت اقدام کاهش می یابد، در حالی که ارزش یک اختیار فروش با افزایش قیمت اقدام افزایش می یابد.
۳. زمان تا انقضا: هر چه زمان تا انقضای اختیار بیشتر باشد، زمان بیشتری برای حرکت دارایی زیربنایی در جهت مورد نظر وجود دارد و ارزش اختیار بیشتر می شود. با نزدیک شدن به تاریخ انقضا، ارزش آپشن کاهش می یابد.
۴. نوسانات: نوسانات اندازه گیری میزان عدم قطعیت یا ریسک در قیمت دارایی پایه است. نوسانات بیشتر ارزش هر دو آپشن تماس و قرار را افزایش می دهد.
۵. نرخ های بهره: نرخ های بهره بر ارزش آپشن‌ها تأثیر می گذارد زیرا بر هزینه حمل دارایی پایه تأثیر می گذارد. نرخ های بهره بالاتر، هزینه حمل دارایی را افزایش می دهد، که ارزش اختیار خرید را کاهش می دهد و ارزش اختیار فروش را افزایش می دهد.

امروزه، قیمت گذاری اختیار معامله و نوسانات ضمنی همچنان مفاهیم مهمی در امور مالی و معاملات هستند. بسیاری از مدل ها و تکنیک های جدید برای در نظر گرفتن شرایط مختلف بازار و بهبود دقت قیمت گذاری آپشن توسعه یافته اند.

مقاله "آپشن‌های قیمت‌گذاری و محاسبه نوسانات ضمنی با استفاده از شبکه‌های عصبی"، در سال ۲۰۱۹ توسط Shuaiqiang Liu ، Cornelis W. Oosterlee و Sander M. Bohte نوشته شد، در مقاله از الگوریتم های مدل بلک شولز و هستون برای تخمین قیمت آپشن‌ها استفاده شده و سپس داده های تولید شده به شبکه عصبی داده شده تا شاخص نوسان و قیمت آپشن‌ها را یاد بگیرد. این روش از شبکه عصبی مصنوعی (ANN) برای کاهش زمان محاسبات نسبت به مدل‌های قیمت‌گذاری رایج استفاده می‌کند. نتایج نشان می‌دهند که ANN قادر است با دقت بالا تر نسبت به مدل‌های قیمت‌گذاری رایج قیمت آپشن اروپایی را تخمین بزند. علاوه بر این، زمان محاسبات ANN به طور قابل توجهی کمتر از مدل‌های قیمت‌گذاری استاندارد است.

در این گزارش ،الگوریتم ها و روش های ارایه شده در مقاله بررسی شده است. هدف اصلی بررسی عملکرد این الگوریتم ها بوده که از طریق اجرا و شبیه سازی آنها توسط شبکه ANN انجام شده است. نکات اصلی و نتایج به دست آمده در پایان هر فصل ارائه شده است. سرفصل های اصلی عبارتند از

۱. معرفی مدل‌ها
۲. معرفی شبکه عصبی
۳. قیمت گذاری آپشن ها و نوسانات ضمنی با شبکه عصبی
۴. نتایج و بحث

کدها و شبیه سازی های مربوطه اجرا شده و در گیت هاب قرار گرفته است.

فصل ۱ معرفی مدل ها

۱-۱ مدل بلک شولز

مدل بلک شولز یک مدل ریاضی است که برای تخمین ارزش ابزارهای مالی به‌ویژه آپشن‌ها استفاده می‌شود. این ابزار توسط Fischer Black و Myron Scholes در سال ۱۹۷۳ توسعه یافت و به یک ابزار استاندارد در امور مالی تبدیل شد. مدل بلک شولز فرض می‌کند که قیمت دارایی پایه از یک حرکت هندسی براونی (نوعی فرآیند تصادفی که برای مدل‌سازی رفتار تصادفی استفاده می‌شود) پیروی می‌کند، و این آپشن اروپایی است (فقط در زمان انقضا قابل اعمال است). این مدل همچنین فرض می‌کند که هیچ هزینه مبادله‌ای وجود ندارد، هیچ سود سهامی برای سهام پایه پرداخت نمی‌شود و نرخ بهره بدون ریسک ثابت است. این مدل از چندین ورودی استفاده می‌کند، از جمله قیمت فعلی دارایی پایه، قیمت پیشنهادی اختیار معامله، زمان تا انقضا، نوسانات مورد انتظار دارایی پایه، و نرخ بهره بدون ریسک. با این ورودی ها، مدل مقدار نظری آپشن را محاسبه می کند. مدل بلک شولز چندین پیامد مهم برای معاملات اختیار دارد. این روشی را برای محاسبه ارزش منصفانه یک اختیار ارائه می دهد که می تواند با قیمت بازار مقایسه شود تا مشخص شود که آیا یک آپشن بیش از قیمت یا کمتر از قیمت است. همچنین نشان می‌دهد که ارزش یک اختیار معامله تحت تأثیر عوامل متعددی از جمله نوسانات دارایی پایه، زمان انقضا، و سطح نرخ‌های بهره است.

با وجود مفروضات آن، مدل بلک شولز به طور گسترده ای مورد استفاده قرار گرفته است و تأثیر قابل توجهی بر توسعه معاملات آپشن‌ها و تئوری مالی داشته است.

مدل بلک شولز را نیز می توان با استفاده از معادلات دیفرانسیل جزئی (PDE) به دست آورد. PDE که بر قیمت یک اختیار خرید اروپایی تحت مدل بلک شولز حاکم است، یک معادله دیفرانسیل جزئی خطی و مرتبه دوم است که به صورت زیر بیان می شود:

$$\frac{\partial v}{\partial t}+\frac{1}{2}\sigma^2 s^2\frac{\partial^2 v}{\partial s^2}+rs\frac{\partial v}{\partial s}-rv=0$$

که در آن:

V = قیمت آپشن خرید به عنوان تابعی از قیمت دارایی اولیه و زمان است

S = قیمت دارایی پایه

t = زمان

σ = نوسانات دارایی پایه

r = نرخ بهره بدون ریسک

معادله بلک شولز را می توان به صورت تحلیلی با استفاده از روش جداسازی متغیرها برای یک آپشن خرید اروپایی حل کرد. فرمول به شرح زیر است:

$$v(S_t,t) = N(d_1)S_t - N(d_2)Ke^{-r(T-t)}$$

که در آن

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}[log(\frac{S_t}{K}) + (r + \frac{\sigma^2}{2}(T-t))]$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

و **N(.)** تابع توزیع نرمال تجمعی و نیز در ادامه قرار می دهیم

$$\tau = T - t$$

۲-۱ نوسان ضمنی

نوسانات ضمنی معیاری از نوسانات مورد انتظار دارایی پایه، مانند یک سهام یا یک شاخص است که توسط قیمت یک اختیار خرید در آن دارایی مشخص می شود. این یک ورودی کلیدی در مدل‌های قیمت‌گذاری اختیار معامله است، مانند مدل بلک شولز و مدل هستون، و یک مفهوم مهم در معاملات اختیار معامله است. نوسانات ضمنی با عملکرد معکوس از قیمت فعلی بازار یک آپشن برای تعیین سطح نوسانی که قیمت اختیار نظری را با قیمت بازار برابر می کند، به دست می آید. این فرآیند به عنوان نوسانات قیمت اختیار معامله شناخته می شود. سپس از نوسانات ضمنی به عنوان ورودی در مدل‌های قیمت‌گذاری اختیار معامله برای محاسبه قیمت نظری آپشن‌های دیگر در همان دارایی پایه استفاده می‌شود و اغلب به صورت درصد بیان می شود همچنین می توان آن را به عنوان انتظار بازار از میزان تغییرات قیمت آتی دارایی پایه تفسیر کرد. نوسانات ضمنی بالا نشان می دهد که بازار انتظار دارد دارایی پایه نوسان بیشتری داشته باشد، در حالی که نوسان ضمنی پایین نشان می دهد که بازار انتظار دارد دارایی پایه دارای نوسانات کمتری باشد. از آن می توان برای مقایسه نوسانات مورد انتظار در دارایی های مختلف یا در دوره های زمانی مختلف استفاده کرد. از آنجایی که نوسانات ضمنی از قیمت بازار آپشنها ناشی می شود، می تواند تحت تأثیر عواملی مانند عرضه و تقاضا برای آپشن‌ها، تغییرات نرخ بهره و تغییرات در احساسات بازار قرار گیرد. بنابراین، نوسانات ضمنی می‌تواند ابزار مفیدی برای معامله‌گران و سرمایه‌گذاران برای ارزیابی انتظارات بازار از تغییرات آتی قیمت و تصمیم‌گیری آگاهانه در مورد استراتژی‌های معاملات اختیار باشد.

نوسانات ضمنی در معادله بلک شولز:

هنگام استفاده از مدل بلک شولز معامله گران و سرمایه گذاران می توانند قیمت بازار یک آپشن را وارد کرده و نوسانات ضمنی را حل کنند با فرض دانستن قیمت اختیار بازار مشاهده شده v^{mkt} نوسانات

ضمنی σ^* را می توان بدست آورد. نوسانات ضمنی پیش بینی یا تضمینی برای نوسانات آینده نیست بلکه بازتابی از قیمت ها و انتظارات بازار در یک برهه زمانی معین است.

$$BS(\sigma^*, s, k, \tau, r) = v^{mkt} \rightarrow \sigma^*(k, \tau) = BS^{-1}(v^{mkt}, s, k, \tau, r)$$

با توجه به اینکه در معادله بلک شولز نوسانات اختیار ثابت فرض می شود پس بااتخاذ $m = S_t / k$ تا زمان سررسید $\tau = T - t$ نوسانات ضمنی $\sigma^* \in [0, \infty)$ را می توان بدست آورد از آنجایی که راه حل تحلیلی برای حل آن وجود ندارد مقدار σ^* را با روش های عددی تکراری همچون روش نیوتن رافسون یا برنت تعیین می کنند.

۱-۲-۱ روش نیوتن رافسون

$$\sigma_{k+1} = \sigma_k - \frac{v(\sigma_k) - v^{mkt}}{f'(\sigma_k)}$$

۱. باتخمین اولیه برای نوسانات ضمنی شروع میکنیم که با σ_0 نشان داده می شود.
۲. از فرمول قیمت گذاری آپشن بلک شولز برای محاسبه قیمت اختیار خرید نظری بر اساس نوسانات تخمینی اولیه σ_0 استفاده کنید.
۳. قیمت اختیار معامله نظری محاسبه شده را با قیمت مشاهده شده در بازار اختیارمعامله مقایسه کنید خطای قیمت را بعنوان تفاوت بین این دو محاسبه کنید.
۴. مشتق قیمت اختیارمعامله را توجه به نوسانات (بعنوان vega شناخته می شود)محاسبه کنید این مشتق حساسیت قیمت اختیارمعامله را نسبت به تغییرات در نوسانات کمیت میدهد.

$$Vega = S * N(d_1) * sqrt(T)$$

۵. نوسانات تخمین زده شده را بااستفاده از فرمول زیر به روز کنید

$$\sigma\_new = \sigma\_old - (option\ price\ caculate - market\ price)/Vega$$

۶. مراحل۲ تا ۵ را تکرار کنید تا خطای قیمت کوچک یا ناچیز شود. روند تکرار تا زمانی ادامه می بابد که قیمت اختیار معامله شده با قیمت بازار در سطح مورد نظر از دقت مطابقت داشته باشد در آن نقطه مقدار نوسان بدست آمده نوسان ضمنی درنظر گرفته می شود.

۲-۲-۱ روش برنت

روش برنت به عنوان یک الگوریتم بدون مشتق قوی و کارآمد است ترکیبی از روش دوبخشی و روش مقطعی secant و درون یابی درجه دوم معکوس است.
برای حل نوسانات ضمنی درقیمت گذاری آپشن ها بکار می رود هدف یافتن مقدار نوسان ضمنی است که باعث می شود قیمت اختیار معامله محاسبه شده با قیمت مشاهده شده در بازار مطابقت داشته باشد.

۱. با تعیین بازه [a,b] برای محدوده احتمالی متغییر نوسانات ضمنی شروع کنید. بطور معمول a را روی یک مقدار مثبت کوچک مانند ۰.۰۰۱ و b را روی یک مقدار بزرگتر مانند ۰.۳تنظیم می شود.
۲. قیمت اختیار معامله را با استفاده از مدل بلک شولز با مقادیر اولیه a,b محاسبه کنید.
۳. بررسی کنید که آیا قیمت اختیار معادله با قیمت اختیار بازار مشاهده شده در یک tolerance از پیش تعریف شده مطابقت دارد یا خیر، اگر تطابق در محدوده tolerance باشد مقدار فعلی نوسانات ضمنی در نظر گرفته می شود و الگوریتم خاتمه می یابد.

Tolerance: درجه دقت مورد نیاز در یک اندازه گیری یا حدود قابل قبولی از مقادیر که در آن اندازه گیری قابل قبول خواهد بود

I. روش(دوبخشی)تقسیم بندی را اعمال کنید:
۱. محاسبه نقطه میانی بازه c=(a+b)/2
۲. قیمت اختیار معامله را با استفاده از مدل بلک شولز با مقدار c محاسبه کنید
۳. بررسی کنید که آیا قیمت اختیار محاسبه شده با قیمت بازار مشاهده شده درمحدوده tolerance مطابقت دارد یا خیر اگر چنین است c نوسانات ضمنی درنظرگرفته می شود و الگوریتم خاتمه می یابد.
۴. اگر قیمت اختیاری محاسبه شده دارای علامت مشابه با قیمت مشاهده شده در بازار باشد a= c تنظیم کنیم در غیر این صورت b= c

II. اعمال درون یابی معکوس درجه دوم
۱. ضرایب چندجمله ای درون یابی درجه دوم را با استفاده از سه نقطه ((a,f(a)), (b,f(b)) , (c,f(c)) محاسبه کنید جایی که f(x) تفاوت بین قیمت اختیار محاسبه شده و قیمت بازار مشاهده شده است
۲. از چندجمله ای درجه دوم برای تخمین مقدار آزمایشی جدید d استفاده کنید
۳. بررسی کنید که آیا مقدار تخمین زده شده d در بازه [a,b] است و شرایط خاصی را برآورده می کند مانند نزدیکتربودن به c تا نیمه راه بین a,b اگر چنین است قیمت اختیار معامله را با استفاده از مدل بلک شولز با مقدار d محاسبه کنید

۴. اگر قیمت اختیاری محاسبه شده با قیمت بازار مشاهده شده در محدوده tolerance مطابقت داشته باشد d نوسان ضمنی در نظر گرفته می شود و الگوریتم خاتمه می یابد

۵. اگر قیمت اختیاری محاسبه شده دارای علامتی مشابه با قیمت مشاهده شده در بازار باشد a = d تنظیم کنید در غیر این صورت b = d را تنظیم کنید

III. روش Secant را اعمال کنید

۱. از نقاط (c,f(c)), (d,f(d)) برای تقریب نوسانات ضمنی استفاده کنید

۲. بررسی کنید که آیا تقریب در بازه [a,b] است و شرایط لازم را برآورده می کند. اگر چنین است قیمت اختیار معامله را با استفاده از مدل بلک شولز با مقدار تقریب جدید محاسبه کنید

۳. اگر قیمت اختیار محاسبه شده با قیمت بازار مشاهده شده در محدوده tolerance مطابقت داشته باشد تقریب جدید نوسانات ضمنی در نظر گرفته می شود و الگوریتم خاتمه می یابد

۴. اگرقیمت اختیاری محاسبه شده دارای علامت مشابه با قیمت مشاهده شده بازار باشد a= d و در غیر اینصورت b = d تنظیم کنید

بازه [a,b] رابراساس نتایج حاصل از دوبخشی درون یابی درجه دوم معکوس یا گام های قطعی بروز کنید و مراحل بالا را تکرار کنید تازمانی که نوسانات ضمنی در tolerance حوزه موردنظر پیدا شود یا به حداکثر تعداد تکرار برسد

در روش برنت تکرار برای یافتن مقدار آزمایش بعدی نوسانات ضمنی σ_{k+1} به شرح زیر است

$$\sigma_{k+1} = \frac{\sigma_k f(\sigma_{k-1}) f(\sigma_{k-2})}{(f(\sigma_k) - f(\sigma_{k-1}))(f(\sigma_k) - f(\sigma_{k-2}))} + \frac{\sigma_{k-1} f(\sigma_{k-2}) f(\sigma_k)}{(f(\sigma_{k-1}) - f(\sigma_{k-2}))(f(\sigma_{k-1}) - f(\sigma_k))} + \frac{\sigma_{k-2} f(\sigma_{k-1}) f(\sigma_k)}{(f(\sigma_{k-2}) - f(\sigma_{k-1}))(f(\sigma_{k-2}) - f(\sigma_k))}$$

هنگامی که دو تقریب متوالی یکسان هستند $\sigma_{k-1} = \sigma_k$ درجه دوم درونیابی با یک روش بر اساس Secant جایگزین می شود

$$\sigma_{k+1} = \sigma_{k-1} - f(\sigma_{k-1}) \frac{\sigma_{k-1} - \sigma_{k-2}}{f(\sigma_{k-1}) - f(\sigma_{k-2})}$$

$$\sigma_{k+1} = \sigma_k - f(\sigma_k) * \frac{(\sigma_k - \sigma_{k-1})}{f(\sigma_k) - f(\sigma_{k-1})}$$

۳-۱ مدل هستون

مدل هستون توسط استیون هستون در سال ۱۹۹۳ توسعه یافت و به یک جایگزین محبوب برای مدل بلک شولز برای آپشن‌های قیمت گذاری تبدیل شد. مدل هستون فرض می کند که نوسان دارایی پایه تصادفی است، به این معنی که به طور تصادفی در طول زمان تغییر می کند، و قیمت سهام و نوسانات از دو فرآیند تصادفی همبسته پیروی می کنند. این مدل همچنین میانگین برگشت نوسان و حق بیمه ریسک نوسان را در بر می گیرد، که غرامتی است که سرمایه گذاران برای تحمل ریسک نوسان نیاز دارند. این مدل از ورودی‌های مختلفی استفاده می‌کند، از جمله قیمت فعلی دارایی پایه، قیمت پیشنهادی اختیار معامله، زمان انقضا، همبستگی مورد انتظار بین قیمت سهام و نوسانات، میانگین سرعت برگشت نوسانات مورد انتظار و حق بیمه ریسک نوسان. با این ورودی ها، مدل مقدار نظری آپشن را محاسبه می کند. همچنین راهی برای محاسبه نوسانات ضمنی یک آپشن ارائه می‌کند، که سطح نوسانی است که قیمت اختیار نظری را با قیمت بازار برابر می‌کند. مدل هستون چندین پیامد مهم برای معامله اختیار دارد. این روشی را برای محاسبه نوسانات تصادفی دارایی پایه ارائه می دهد، که می تواند تأثیر قابل توجهی بر قیمت های اختیار داشته باشد، به ویژه برای آپشن‌های با تاریخ طولانی تر. همچنین نشان می دهد که نوسانات دارایی پایه یک عامل کلیدی در قیمت گذاری آپشن‌ها است و در طول زمان ثابت نیست. با وجود پیچیدگی، مدل هستون به طور گسترده ای مورد استفاده قرار گرفته است و تأثیر قابل توجهی در توسعه معاملات آپشن‌ها و تئوری مالی داشته است.

مدل هستون توسط سیستم دو متغیره معادلات دیفرانسیل تصادفی به فرم زیر است:

$$dS_t = rS_t dt + \sqrt{v_t} S_t dW_{1,t}$$
$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v_t} dW_{2,t}$$
$$E^p[dW_{1,t}, dW_{2,t}] = \rho dt$$

برای سادگی اندیس زمان را ساده می کنیم و $S = S_t, v = v_t, W_1 = W_{1,t}, W_2 = W_{2,t}$

پارامترهای مدل بصورت زیر تعریف می شود

r: رانش فرآیند برای سهام

$\kappa \succ 0$: میانگین سرعت برگشت برای واریانس(سرعت بازگشت میانگین)

θ: میانگین بلندمدت نوسان

σ: نوسان برای واریانس (نوسان نوسانات)

v_0: مقدار اولیه برای واریانس

ρ: کورولیشن بین دو حرکت براونی W_1, W_2

فرمول مدل هستون برای آپشن اروپایی با معادله دیفرانسیل جزئی (PDE) مربوطه به صورت زیر بیان می شود:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \kappa(\theta - v)\frac{\partial V}{\partial v} + \frac{1}{2}vS^2\frac{\partial^2 V}{\partial S^2} + \rho\sigma vS\frac{\partial^2 V}{\partial S \partial v} + \frac{1}{2}v\sigma^2\frac{\partial^2 V}{\partial v^2} = 0$$

توجه به این نکته مهم است که PDE قیمت گذاری آپشن هستون دارای راه حل تحلیلی نیست. بنابراین روش های تقریب عددی معمولا برای حل PDE و بدست آوردن قیمت آپشن ها استفاده می شود. چندین روش برای حل PDE وجود دارد، از جمله:

۱. روش های تفاضل محدود: این روش ها، PDE هستون را به سیستمی از معادلات تفاضل محدود که می توان به صورت عددی حل کرد، گسسته می کند. رایج‌ترین روش‌های تفاضل محدود مورد استفاده برای حل PDE هستون، روش کرانک-نیکولسون است.

۲. روش های تبدیل فوریه: این روش ها از تبدیل فوریه برای تبدیل PDE هستون به یک معادله جبری استفاده می کنند که می تواند به صورت تحلیلی حل شود. متداول‌ترین روش‌های تبدیل فوریه که برای حل PDE هستون استفاده می‌شود، روش Carr-Madan و روش Lewis و روش فوریه کسینوس است.

۳. روش های شبیه سازی مونت کارلو: این روش ها فرآیند تصادفی زیربنای مدل هستون را با استفاده از شبیه سازی مونت کارلو شبیه سازی می کنند. سپس قیمت اختیار معامله به عنوان میانگین تنزیل شده بازده های شبیه سازی شده تخمین زده می شود. متداول‌ترین روش‌های شبیه‌سازی مونت کارلو برای حل PDE هستون، روش اویلر-مارویاما و روش میلشتاین هستند.

در این مقاله در طول آموزش Heston-ANN از مدل هستون با روش کسینوسی استفاده شده است. برای مدل هستون، معادله قیمت گذاری COS نیز ساده شده است،

$$v(\mathbf{x}, t_0, u_0) \approx \mathbf{K}e^{-r\Delta t} \cdot \mathrm{Re}\left\{\sum{}'_{k=0}^{N-1} \varphi_{hes}\left(\frac{k\pi}{b-a}; u_0\right) U_k \cdot e^{ik\pi\frac{\mathbf{x}-n}{b-a}}\right\}.$$

که در آن Σ' نشان می دهد که اولین جمله در جمع با یک دوم، وزن می شود و تابع مشخصه $\varphi_{hes}(\omega; u_0)$ بصورت زیر محاسبه می‌شود:

$$\varphi_{hes}(\omega;u_0) = \exp\left(i\omega\mu\Delta t + \frac{u_0}{\eta^2}\left(\frac{1-e^{-D\Delta t}}{1-Ge^{-D\Delta t}}\right)(\lambda - i\rho\eta\omega - D)\right)$$
$$\cdot \exp\left(\frac{\lambda\bar{u}}{\eta^2}\left(\Delta t(\lambda - i\rho\eta\omega - D) - 2\log\left(\frac{1-Ge^{-D\Delta t}}{1-G}\right)\right)\right),$$

$$D = \sqrt{(\lambda - i\rho\eta\omega)^2 + (\omega^2 + i\omega)\eta^2} \text{ and } G = \frac{\lambda - i\rho\eta\omega - D}{\lambda - i\rho\eta\omega + D}.$$

که در آن μ=r و داریم

$$U_k = \begin{cases} \frac{2}{b-a}(\chi_k(0,b) - \psi_k(0,b)) & \text{for a call,} \\ \frac{2}{b-a}(-\chi_k(a,0) + \psi_k(a,0)) & \text{for a put.} \end{cases}$$

و

$$\chi_k(c,d) := \frac{1}{1+\left(\frac{k\pi}{b-a}\right)^2}\left[\cos\left(k\pi\frac{d-a}{b-a}\right)e^d - \cos\left(k\pi\frac{c-a}{b-a}\right)e^c + \frac{k\pi}{b-a}\sin\left(k\pi\frac{d-a}{b-a}\right)e^d - \frac{k\pi}{b-a}\sin\left(k\pi\frac{c-a}{b-a}\right)e^c\right]$$

$$\psi_k(c,d) := \begin{cases} \left[\sin\left(k\pi\frac{d-a}{b-a}\right) - \sin\left(k\pi\frac{c-a}{b-a}\right)\right]\frac{b-a}{k\pi}, & k \neq 0, \\ (d-c), & k = 0. \end{cases}$$

نحوه تعیین بازه [a, b] بصورت زیر است،

$$[a, b] := [\, c_1 - L\sqrt{|c_2|},\ c_1 + L\sqrt{|c_2|}\,]$$

$$c1 = r\tau + (1 - e^{-\kappa\tau})\frac{\theta - v0}{2\kappa} - 0.5\theta\tau$$

$$c2 = \frac{1}{8\kappa^3}[\sigma\tau\kappa e^{-\kappa\tau}(v0-\theta)(8\kappa\rho - 4\sigma) + \kappa\rho\sigma(1-e^{-\kappa\tau})(16\theta - 8v0) + 2\theta\kappa\tau(-4\kappa * \rho\sigma + \sigma^2 + 4\kappa^2) + \sigma^2((\theta - 2v0)\,e^{-2\kappa\tau}) + 8\kappa^2(v0-\theta)(1-e^{-\kappa\tau})] + \theta(6e^{-\kappa\tau} - 7) + 2v0)$$

فصل ۲ معرفی شبکه عصبی

۲-۱ شبکه عصبی ANN

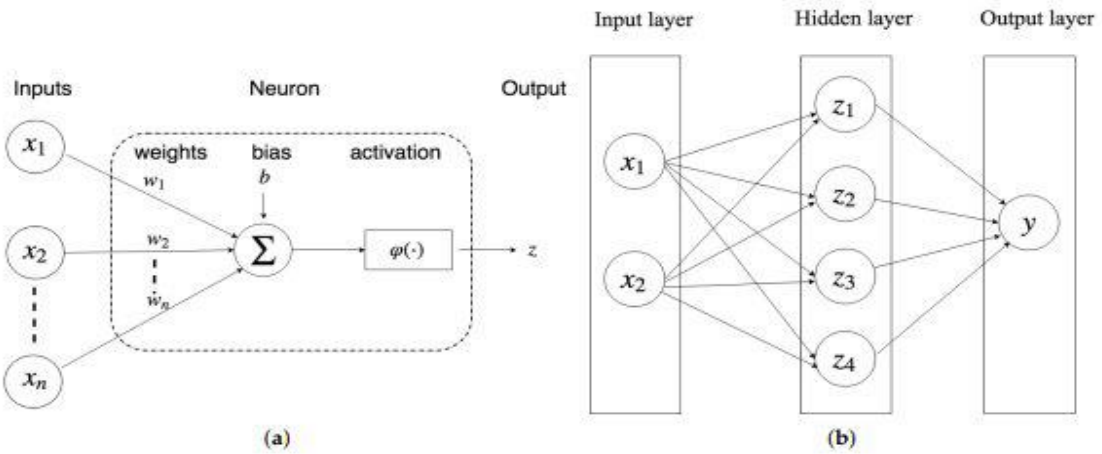
شبکه عصبی مصنوعی (ANN) نوعی مدل یادگیری ماشینی است که به طور آزاد بر اساس ساختار و عملکرد مغز انسان است. این یک مدل محاسباتی است که از گره های(nodes) به هم پیوسته (که نورون ها یا واحدها نیز نامیده می شوند) تشکیل شده است که در لایه ها سازماندهی شده اند. ANN ها برای یادگیری از داده ها با تنظیم نقاط قوت اتصالات بین گره ها بر اساس الگوهای موجود در داده های ورودی طراحی شده اند. داده های ورودی از طریق لایه ورودی وارد شبکه می شود و خروجی شبکه از لایه خروجی تولید می شود. لایه‌های پنهان بین لایه‌های ورودی و خروجی، یک سری محاسبات را روی داده‌های ورودی انجام می‌دهند تا آن‌ها را به شکلی تبدیل کنند که برای خروجی نهایی مفید باشد. رایج ترین نوع ANN شبکه عصبی پیشخور است، به این معنی که اطلاعات تنها در یک جهت، از لایه ورودی به لایه خروجی، بدون هیچ گونه حلقه بازخوردی جریان می یابد. گره های هر لایه از طریق وزنه هایی به گره های لایه های مجاور متصل می شوند که در حین تمرین تنظیم می شوند تا خطا بین خروجی پیش بینی شده و خروجی واقعی به حداقل برسد.

آموزش ANN شامل تغذیه مقدار زیادی از داده های برچسب گذاری شده و تنظیم وزن ها بر اساس خطای بین خروجی پیش بینی شده و خروجی واقعی است. این فرآیند معمولاً با استفاده از الگوریتمی به نام پس انتشار انجام می شود که گرادیان خطا را نسبت به وزن ها محاسبه می کند و سپس وزن ها را در جهتی به روز می کند که خطا را به حداقل می رساند.

شبکه‌های عصبی مصنوعی در طیف وسیعی از کاربردها، از جمله تشخیص تصویر و گفتار، پردازش زبان طبیعی و مدل‌سازی پیش‌بینی‌کننده در امور مالی استفاده شده‌اند. آنها به ویژه در یادگیری الگوها و روابط پیچیده در داده ها موثر هستند و اغلب در موقعیت هایی که مدل های آماری سنتی مناسب نیستند استفاده می شوند.

روش شناسی

شکل ۱



در این بخش، شبکه عصبی برای تقریب تابع مدل‌های مالی ارائه می‌کنیم.

این روش شامل دو جزء اصلی است، تولید کننده برای ایجاد داده های مالی برای آموزش مدل و پیش بینی کننده (ANN) برای تقریب قیمت های اختیار معامله بر اساس مدل آموزش دیده. چارچوب داده محور شامل مراحل زیر است (الگوریتم ۱)،

شکل ۲

Algorithm 1 Model framework

1: – Generate the sample data points for input parameters,
2: – Calculate the corresponding output (option price or implied volatility) to form a complete data set with inputs and outputs,
3: – Split the above data set into a training and a test part,
4: – Train the ANN on the training data set,
5: – Evaluate the ANN on the test data set,
6: – Replace the original solver by the trained ANN in applications.

الگوریتم ۱:

۱. ایجاد نقاط داده نمونه برای پارامترهای ورودی،
۲. محاسبه خروجی مربوطه (قیمت گزینه یا نوسانات ضمنی) برای تشکیل یک داده کامل مجموعه با ورودی و خروجی،
۳. مجموعه داده های فوق را به بخش آموزشی و آزمایشی تقسیم کنید.

۴. آموزش ANN در مجموعه داده های آموزشی،
۵. ANN را در مجموعه داده های آزمایشی ارزیابی کنید،
۶. حل کننده اصلی را با ANN آموزش دیده در برنامه ها جایگزین کنید.

اکنون به بررسی چند پارامتر مهم که در ساخت شبکه ANN به آنها نیاز داریم می‌پردازیم:

- فعال سازی (Activation)

در زمینه شبکه های عصبی مصنوعی، فعال سازی به تابع ریاضی اطلاق می شود که بر خروجی هر نورون در یک شبکه عصبی اعمال می شود. تابع فعال سازی به معرفی غیر خطی بودن به شبکه کمک می کند، که برای توانایی شبکه در یادگیری الگوها و روابط پیچیده در داده ها مهم است. تابع فعال سازی مجموع وزنی ورودی ها را به نورون می گیرد و یک تبدیل غیرخطی را برای تولید خروجی نورون اعمال می کند. سپس خروجی تابع فعال سازی به لایه بعدی نورون ها در شبکه منتقل می شود. انواع مختلفی از توابع فعال سازی وجود دارد که می توانند در شبکه های عصبی استفاده شوند، از جمله sigmoid، tanh، ReLU ، Leaky ReLU و غیره. انتخاب تابع فعال‌سازی می‌تواند تأثیر قابل‌توجهی بر عملکرد شبکه داشته باشد و عملکردهای مختلف فعال‌سازی ممکن است کم و بیش برای انواع داده‌ها و وظایف مناسب باشند.

ReLU (Rectified Linear Unit) یک تابع فعال سازی است که معمولاً در شبکه های عصبی مصنوعی استفاده می شود. این یک تابع ساده و کارآمد محاسباتی است که نشان داده شده است که در انواع مختلف شبکه های عصبی به خوبی کار می کند. تابع ReLU مجموع وزنی ورودی‌ها را به یک نورون می‌گیرد و یک تبدیل غیرخطی را برای تولید خروجی نورون اعمال می‌کند. به طور خاص، تابع ReLU در صورت مثبت بودن ورودی، و در غیر این صورت ۰ را برمی‌گرداند. این بدان معناست که نورون های ReLU فقط زمانی فعال هستند که ورودی آنها مثبت باشد و در غیر این صورت به خروجی شبکه کمکی نمی کنند. از مزایای استفاده از ReLU می توان به سادگی، کارایی محاسباتی و اثربخشی آن در جلوگیری از مشکل ناپدیدشدن گرادیان اشاره کرد که می تواند هنگام استفاده از سایر توابع فعال سازی رخ دهد. علاوه بر این، نشان داده شده است که ReLU در شبکه های عصبی عمیق، که شبکه هایی با لایه های متعدد هستند، به خوبی کار می کند. با این حال، یکی از معایب احتمالی ReLU این است که می‌تواند به نورون‌های «مرده» منجر شود، که نورون‌هایی هستند که همیشه خروجی صفر دارند و به خروجی شبکه کمک نمی‌کنند. این می تواند زمانی اتفاق بیفتد که وزن یک نورون به گونه ای باشد که ورودی تابع ReLU همیشه منفی باشد. برای پرداختن به این مسئله، انواع ReLU مانند Leaky ReLU و Parametric ReLU پیشنهاد شده است که می تواند به جلوگیری از مرگ نورون ها کمک کند.

- نرخ انصراف (Dropout rate)

نرخ رها شدن یا نرخ خروج (نرخ انصراف) یک فراپارامتر(یا هایپر پارامتر) در شبکه های عصبی مصنوعی است که احتمال حذف یا "خاموش" شدن نورون ها را در طول آموزش کنترل می کند. Dropout یک تکنیک منظم سازی است که به جلوگیری از برازش بیش از حد و بهبود عملکرد تعمیم شبکه کمک می کند. در طول آموزش، هر نورون در یک لایه با احتمال مشخصی از بین می‌رود، که توسط فراپارامتر نرخ خروج کنترل می‌شود. این بدان معنی است که خروجی هر نورون با احتمال مشخصی روی ۰ تنظیم می شود و به طور موثر آن نورون را از شبکه حذف می کند و از مشارکت آن در خروجی جلوگیری می کند. سپس نورون‌های باقی‌مانده در لایه با یک عامل، بزرگ‌تر می‌شوند تا بزرگی کلی خروجی حفظ شود. نرخ انصراف معمولاً روی مقداری بین ۰ و ۱ تنظیم می شود، که در آن ۰ به معنای عدم Dropout یعنی همه نورون ها فعال هستند و ۱ به معنای انصراف کامل (یعنی همه نورون ها خاموش هستند) است. یک مقدار رایج برای نرخ انصراف ۰.۵ است، به این معنی که هر نورون ۵۰٪ احتمال دارد که در طول تمرین حذف شود. نرخ انصراف یک فراپارامتر است که باید در طول فرآیند آموزش تنظیم شود تا بهترین عملکرد شبکه در مجموعه اعتبار سنجی به دست آید. نرخ بهینه انصراف می تواند بسته به اندازه و پیچیدگی شبکه، ماهیت داده ها و عوامل دیگر متفاوت باشد.

- نرمال سازی دسته ای (Batch-normalization)

نرمال سازی دسته ای تکنیکی در شبکه های عصبی مصنوعی است که به بهبود عملکرد آموزشی و تعمیم شبکه کمک می کند. این شامل عادی سازی فعال سازی هر لایه در شبکه با توجه به آمار کوچک دسته ای از نمونه های در حال پردازش است. در طول آموزش، ورودی به هر لایه از شبکه با کم کردن میانگین و تقسیم بر انحراف استاندارد دسته کوچک، نرمال می شود. این به کاهش تغییر متغیر داخلی کمک می کند، که پدیده ای است که در آن توزیع فعال سازی ها در هر لایه از شبکه با به روز شدن پارامترهای شبکه در طول آموزش تغییر می کند. با عادی سازی فعال سازی هر لایه، نرمال سازی دسته ای به تثبیت فرآیند آموزش و بهبود نرخ همگرایی شبکه کمک می کند. نرمال سازی دسته ای همچنین می تواند به عنوان نوعی منظم سازی عمل کند، زیرا نویز را با عادی سازی هر مینی دسته به طور مستقل وارد شبکه می کند. این می تواند به جلوگیری از تعمیم بیش از حد و بهبود عملکرد تعمیم شبکه کمک کند. نرمال سازی دسته ای را می توان برای انواع مختلفی از لایه ها در شبکه، از جمله لایه های کاملا متصل، لایه های کانولوشن و لایه های تکراری اعمال کرد. این به یک تکنیک رایج در یادگیری عمیق تبدیل شده است و اغلب در مدل های پیشرفته برای کارهای مختلف استفاده می شود. یک اشکال بالقوه نرمال سازی دسته ای این است که می تواند هزینه محاسباتی آموزش را افزایش دهد، زیرا برای محاسبه میانگین و واریانس هر مینی دسته به محاسبات اضافی نیاز دارد. با این حال، این می تواند با استفاده از پیاده سازی های بهینه و شتاب سخت افزاری کاهش یابد.

- مقداردهی اولیه (Initialization)

مقداردهی اولیه در شبکه های عصبی مصنوعی به فرآیند تنظیم مقادیر اولیه وزن ها و بایاس ها در شبکه قبل از شروع آموزش اشاره دارد. مقداردهی اولیه مناسب پارامترها برای همگرایی و عملکرد شبکه در طول آموزش مهم است. مقادیر اولیه وزن ها و بایاس ها می تواند به طور قابل توجهی بر روند آموزش

و عملکرد نهایی شبکه تأثیر بگذارد. اگر وزن‌ها خیلی کوچک اولیه شوند، ممکن است فعال‌سازی‌ها بسیار کوچک شوند و منجر به مشکل ناپدیدشدن گرادیان شوند، جایی که گرادیان‌ها بسیار کوچک می‌شوند و شبکه قادر به یادگیری نیست. از طرف دیگر، اگر وزن‌ها خیلی بزرگ اولیه شوند، ممکن است فعال‌سازی‌ها بسیار بزرگ شوند و منجر به مشکل گرادیان انفجاری شوند، جایی که گرادیان‌ها بسیار بزرگ می‌شوند و شبکه ناپایدار می‌شود. روش‌های اولیه‌سازی مختلفی برای کاهش این مسائل و کمک به همگرایی سریع‌تر شبکه و دستیابی به عملکرد بهتر پیشنهاد شده‌اند. برخی از روش های اولیه سازی رایج عبارتند از:

۱. مقداردهی اولیه تصادفی: در این روش وزن ها و بایاس ها به صورت تصادفی با استفاده از توزیع یکنواخت یا نرمال مقداردهی اولیه می شوند. این روش می تواند برای شبکه های کم عمق به خوبی کار کند، اما ممکن است برای شبکه های عمیق تر بهینه نباشد.

۲. مقدار دهی اولیه Xavier: این روش واریانس وزن ها را با مقداری تنظیم می کند که با تعداد ورودی های نورون نسبت معکوس دارد. این کمک می کند تا فعال سازی ها در یک محدوده معقول نگه داشته شوند و از مشکل گرادیان ناپدید شدن یا انفجار جلوگیری شود.

۳. مقداردهی اولیه He: این روش واریانس وزن ها را به مقداری تنظیم می کند که با تعداد ورودی های نورون نسبت معکوس دارد، اما در ۲ ضرب می شود. این به محاسبه تابع فعال سازی ReLU کمک می کند که می تواند منجر به فعال سازی های بزرگتر از سایر توابع فعال سازی شود.

انتخاب روش اولیه می تواند به معماری شبکه، تابع فعال سازی استفاده شده و ماهیت داده ها بستگی داشته باشد. مقداردهی اولیه مناسب می تواند به همگرایی سریعتر شبکه و دستیابی به عملکرد بهتر کمک کند.

- بهینه‌ساز (optimizer)

در شبکه‌های عصبی مصنوعی، بهینه‌ساز الگوریتمی است که برای به‌روزرسانی وزن‌ها و بایاس‌های شبکه در حین آموزش، با هدف به حداقل رساندن عملکرد تلفات و بهبود عملکرد شبکه استفاده می‌شود. بهینه ساز با محاسبه گرادیان های تابع تلفات با توجه به وزن ها و بایاس ها کار می کند و از این گرادیان ها برای به روز رسانی پارامترها به گونه ای استفاده می کند که شبکه را به سمت تلفات کمتری حرکت می دهد. انتخاب بهینه ساز می تواند تاثیر بسزایی در عملکرد آموزشی و عملکرد نهایی شبکه داشته باشد. انواع مختلفی از بهینه سازها پیشنهاد شده اند که هر کدام نقاط قوت و ضعف خاص خود را دارند. برخی از بهینه سازهای رایج عبارتند از:

۱. Stochastic Gradient Descent (SGD): این یک بهینه ساز ساده و پرکاربرد است که وزن ها را با برداشتن یک گام کوچک در جهت گرادیان منفی تابع ضرر به روز می کند.

۲. Adam: این یک بهینه ساز تطبیقی است که از ترکیب لحظه های اول و دوم گرادیان برای تنظیم نرخ یادگیری برای هر وزن استفاده می کند که می تواند به بهبود همگرایی و عملکرد کمک کند.

۳. RMSProp: این بهینه ساز تطبیقی دیگری است که از میانگین متحرک گرادیان های مجذور

برای تنظیم نرخ یادگیری برای هر وزن استفاده می کند، که می تواند به جلوگیری از بزرگ شدن نرخ یادگیری کمک کند.

۴. AdaGrad: این بهینه ساز نرخ یادگیری را برای هر وزن بر اساس اطلاعات گرادیان تاریخی برای آن وزن تطبیق می دهد، که می تواند به پارامترهایی که دارای مقدار بروزرسانی کمی هستند وزن بیشتری بدهد.

انتخاب بهینه ساز می تواند به عوامل مختلفی مانند اندازه و پیچیدگی شبکه، ماهیت داده ها و معیارهای عملکرد مورد علاقه بستگی داشته باشد. اغلب لازم است با بهینه سازها و هایپرپارامترهای مختلف آزمایش شود تا بهترین ترکیب را برای یک کار مشخص پیدا کنید.

- اندازه دسته‌ای (Batch size)

در شبکه‌های عصبی مصنوعی، اندازه دسته‌ای به تعداد نمونه‌های آموزشی اطلاق می‌شود که در طی آموزش در یک گذر به جلو/عقب شبکه پردازش می‌شوند. انتخاب اندازه دسته ای می تواند تاثیر قابل توجهی بر عملکرد آموزشی و همگرایی شبکه داشته باشد. اندازه دسته‌ای کوچک به این معنی است که شبکه به‌طور مکرر به‌روزرسانی می‌شود و به‌روزرسانی‌ها براساس حجم کمتری از داده‌ها است. این می تواند منجر به همگرایی سریعتر و نیازهای کمتر حافظه شود، زیرا در هر زمان داده های کمتری باید در حافظه ذخیره شود. با این حال، اندازه‌های دسته‌ای کوچک نیز می‌توانند منجر به به‌روزرسانی‌های پر سر و صدا و هم‌گرایی کندتر شوند، زیرا به‌روزرسانی‌ها بر اساس داده‌های کمتری هستند و ممکن است شیب واقعی عملکرد از دست دادن را دریافت نکنند. اندازه دسته بزرگ به این معنی است که شبکه با دفعات کمتری به روز می شود و به روز رسانی ها بر اساس حجم بیشتری از داده ها است. این می‌تواند منجر به به‌روزرسانی‌های دقیق‌تر و هم‌گرایی سریع‌تر شود، زیرا به‌روزرسانی‌ها بر اساس داده‌های بیشتری هستند و ممکن است گرادیان‌های واقعی تابع ضرر را بهتر به تصویر بکشند. با این حال، اندازه‌های دسته‌ای بزرگ نیز می‌توانند به حافظه و منابع محاسباتی بیشتری نیاز داشته باشند، زیرا داده‌های بیشتری باید در حافظه ذخیره شده و در هر پاس رو به جلو/عقب پردازش شود. انتخاب اندازه دسته می تواند به عوامل مختلفی مانند اندازه و پیچیدگی شبکه، ماهیت داده ها و منابع محاسباتی موجود بستگی داشته باشد. اغلب لازم است با اندازه های مختلف دسته ای آزمایش شود تا انتخاب بهینه برای یک کار مشخص شود. اندازه های معمول دسته ای از چند ده تا چند صد یا حتی هزاران نمونه در هر دسته متغیر است.

- نرخ یادگیری (learning rate)

در شبکه های عصبی مصنوعی، نرخ یادگیری یک فراپارامتر است که اندازه گام برداشته شده در طول هر بروزرسانی پارامترهای شبکه (وزن ها و سوگیری ها) را در طول آموزش کنترل می کند. نرخ یادگیری تعیین می کند که شبکه با چه سرعتی به یک راه حل بهینه همگرا می شود. اگر نرخ یادگیری خیلی زیاد باشد، شبکه ممکن است از راه حل بهینه عبور کند و واگرا شود و در نتیجه عملکرد ناپایدار یا ضعیفی داشته باشد. اگر نرخ یادگیری خیلی کم باشد، شبکه ممکن است بسیار کند همگرا شود یا در حداقل محلی گیر کند و در نتیجه عملکردی کمتر از حد مطلوب داشته باشد.

برخی از استراتژی های رایج برای انتخاب نرخ یادگیری عبارتند از:

۱. نرخ یادگیری ثابت (Fixed learning rate): یک نرخ یادگیری ثابت انتخاب شده و در طول فرآیند آموزش استفاده می شود. این استراتژی ساده و آسان برای پیاده سازی است، اما ممکن است برای همه شبکه ها و مجموعه داده ها مطلوب نباشد.
۲. کاهش نرخ یادگیری (Learning rate decay): میزان یادگیری به تدریج در طول زمان کاهش می یابد، یا با یک عامل ثابت یا با استفاده از برنامه پیچیده تر. این استراتژی می تواند به جلوگیری از بیش از حد شبکه از راه حل بهینه و بهبود همگرایی کمک کند.
۳. نرخ یادگیری تطبیقی (Adaptive learning rate): نرخ یادگیری به صورت پویا بر اساس گرادیان های تابع ضرر و/یا عوامل دیگر تنظیم می شود. نمونه هایی از روش های نرخ یادگیری تطبیقی عبارتند از AdaGrad، RMSProp و Adam.

انتخاب نرخ یادگیری می تواند به عوامل مختلفی مانند اندازه و پیچیدگی شبکه، ماهیت داده ها و الگوریتم بهینه سازی مورد استفاده بستگی داشته باشد. برای یافتن انتخاب بهینه برای یک کار معین، اغلب لازم است با نرخ ها و استراتژی های مختلف یادگیری آزمایش شود. یک رویکرد رایج استفاده از مجموعه اعتبارسنجی برای نظارت بر عملکرد شبکه در طول آموزش و تنظیم نرخ یادگیری بر این اساس است.

۲-۲ پیکربندی هایپرپارامتر برای شبکه عصبی

همانطور که در جدول ۱ نشان داده شده است، هر مدل ۲۰۰ ایپاک (یک ایپاک زمانی است که مدل کل مجموعه داده های آموزشی را پردازش کرده است) با استفاده از MSE به عنوان معیار ضرر آموزش داده شده است. مشخص شد که دقت پیش‌بینی با افزایش اندازه مجموعه داده‌های آموزشی افزایش می‌یابد. بنابراین، جستجوی تصادفی بر روی یک مجموعه داده کوچک اجرا می‌شود، که سپس با آموزش ANN انتخاب شده بر روی مجموعه داده‌های بزرگتر دنبال می‌شود.

جدول ۱

Parameters	**Options or Range**
Activation	ReLU, Tanh, Sigmoid, ELU
Dropout rate	[0.0, 0.2]
Neurons	[200, 600]
Initialization	uniform, Glorot_uniform, He_uniform
Batch normalization	yes, no
Optimizer	SGD, RMSprop, Adam
Batch size	[256, 3000]

پیکربندی‌های شبکه با عملکرد بهتر با تغییر مقادیر پارامترهای چندگانه آنها برای به دست آوردن مدل ANN نهایی در جدول ۲ ذکر شده است، در مقایسه با Sigmoid، ReLU احتمالاً منجر به همگرایی بهتری می شود و به عنوان یک افزونه به SGD، آدام می تواند مشکل بهینه سازی را به شیوه ای سازگارتر مدیریت کند. با این حال، نرمال سازی دسته ای و حذف کردن دقت مدل را در این مسئله رگرسیونی بهبود نمی بخشد و یک دلیل احتمالی این است که خروجی به پارامترهای ورودی حساس است. متعاقباً، شبکه انتخاب شده را بر روی کل مجموعه داده (آموزش و اعتبارسنجی) آموزش می دهیم

تا وزن های نهایی را به دست آوریم. این رویکرد می تواند به یک ANN با دقت کافی برای تقریب مقادیر قیمت گذاری گزینه منجر شود.

جدول ۲

Parameter	Options
Hidden layers	4
Neurons (each layer)	400
Activation function	ReLU
Dropout rate	0.0
Batch normalization	No
Initialization	Glorot_uniform
Optimizer	Adam
Batch size	1024

۳-۲ معیارهای ارزیابی

ما عملکرد شبکه‌های عصبی مصنوعی را برای حل مدل‌های مالی براساس موارد زیر نشان می‌دهیم. مقادیر واقعی و پیش بینی شده را برای محاسبه دقت یک مدل رگرسیون مقایسه می‌کنیم. معیارهای ارزیابی نقش کلیدی را در توسعه یک مدل ایفا می‌کنند، زیرا بینشی را در زمینه‌هایی که نیاز ببه بهبود دارند ارائه می‌دهند. چند مورد ار معیارهایی که در این گزارش کار استفاده کردیم عبارتند از:

۱. MAE: مخفف Mean Absolute Error است که میانگین اختلاف مطلق بین مقادیر پیش بینی شده و مقادیر واقعی را اندازه‌گیری می‌کند. هدف به حداقل رساندن MAE به منظور پیش بینی بهتر است و نسبت به شاخص‌های پرت، قوی تر از معیار MSE است، که تمایل به حساس بودن به نقاط پرت دارد

$$MAE = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|$$

۲. MSE: مخفف Mean Squared Error است که میانگین اختلاف مجذور بین مقادیر پیش بینی شده و مقادیر واقعی را اندازه گیری می‌کند. هدف به حداقل رساندن MSE به منظور پیش بینی بهتر است.

$$MSE = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2$$

۳. R2-squared: یک خطا نیست، بلکه یک معیار محبوب برای اندازه‌گیری عملکرد مدل است که نشان دهنده نزدیکی نقاط داده به خط رگرسیون برازش است. هر چه مقدار آ« بالاتر باشد، مدل بهتر با داده‌ها مطابقت دارد. بهترین مقدار ممکن ۱ است و می‌تواند منفی نیز باشد(زیرا مدل می‌تواند خودسرانه بدتر باشد).

$$R2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}},$$
$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}.$$

فصل۳ قیمت گذاری آپشن‌ها و نوسانات ضمنی با شبکه عصبی

کدهای این بخش در گیت هاب قرار گرفته شده است و فقط به بیان توضیحات می‌پردازیم.
اصول کلی بدین صورت است که ابتدا داده‌ها توسط مدل‌ها ساخته می‌شوند و از آنها به عنوان ورودی به شبکه‌ای که در بخش قبل معرفی کرده‌ایم استفاده می‌کنیم و درنهایت به ارزیابی آنها می‌پردازیم.

شبکه عصبی قیمت گذاری آپشن و نوسان ضمنی بلک شولز

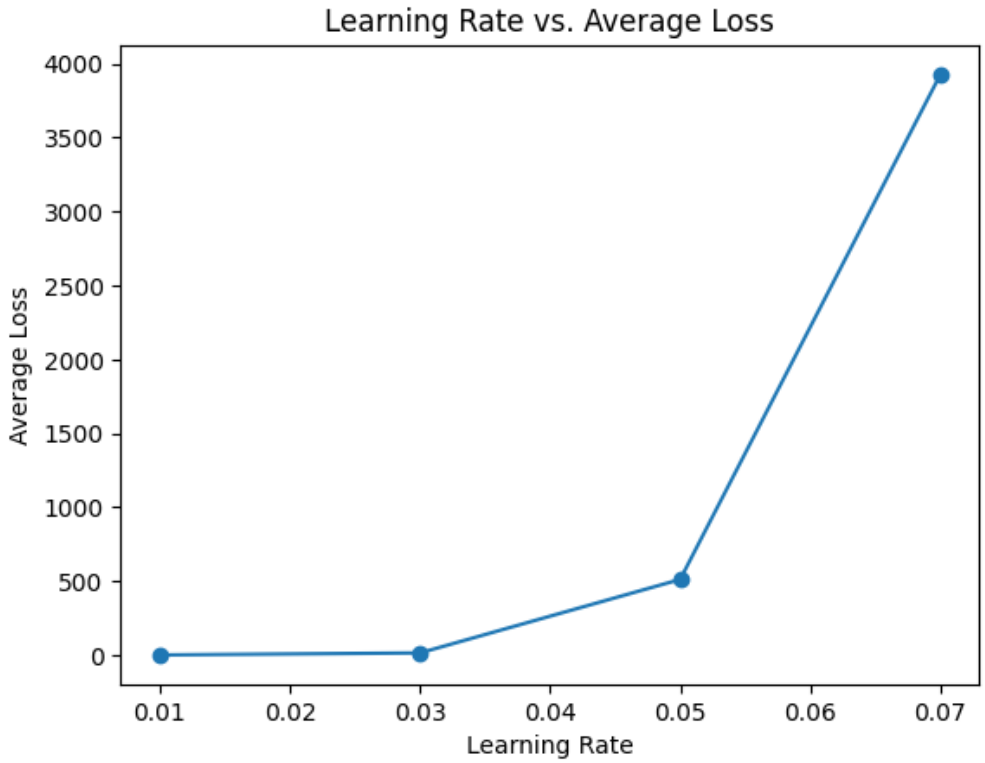
ابتدا داده‌ها را توسط مدل بلک شولز با مقادیری که در جدول۳ نشان داده‌ایم می‌سازیم، کد مربوطه با نام BS_Data_Generation قرار گرفته شده است.

جدول ۳

BS-ANN Parameters	Range	Unit
Input		
Stock price S0/K (SK)	[0.5, 1.5]	-
Time to Maturity τ (T)	[0.3, 0.95]	year
Risk free rate (r)	[0.03, 0.08]	-
Volatility σ (sigma)	[0.02, 0.9]	-
Output		
Call price V/K (Price)	(0.0, 0.73)	-

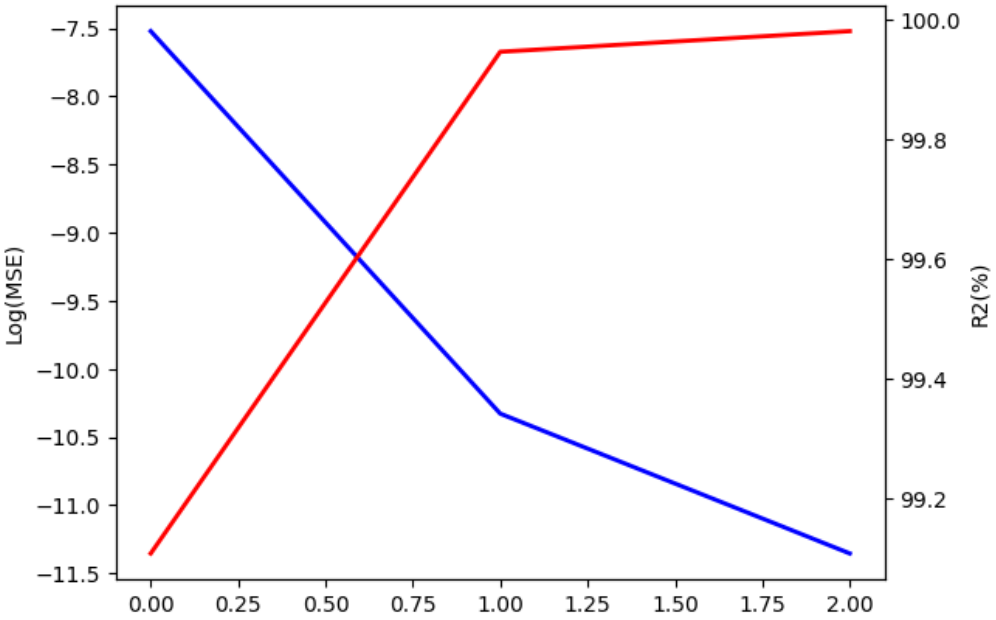
در گام بعد داده‌ها را به داده‌های آموزش و تست که نسبت آنها به ترتیب ۰.۹ به ۰.۱ است تقسیم می‌شوند. و شبکه را با داده‌های آموزشی، آموزش میبینند و با کمک معیارهایی که در بخش قبل توضیح داده شد به کمک داده‌های تست ارزیابی می‌شوند. در این گزارش کار به دلیل عدم دسترسی به سرور برای اجرا تعداد ایپاک کم استفاده شد ولی نتیجه همچنان با دقت بالایی بدست آمده که در فایل مربوط به بلک شولز نتایج به همراه نمودارها قرار گرفته شد.

شکل ۳



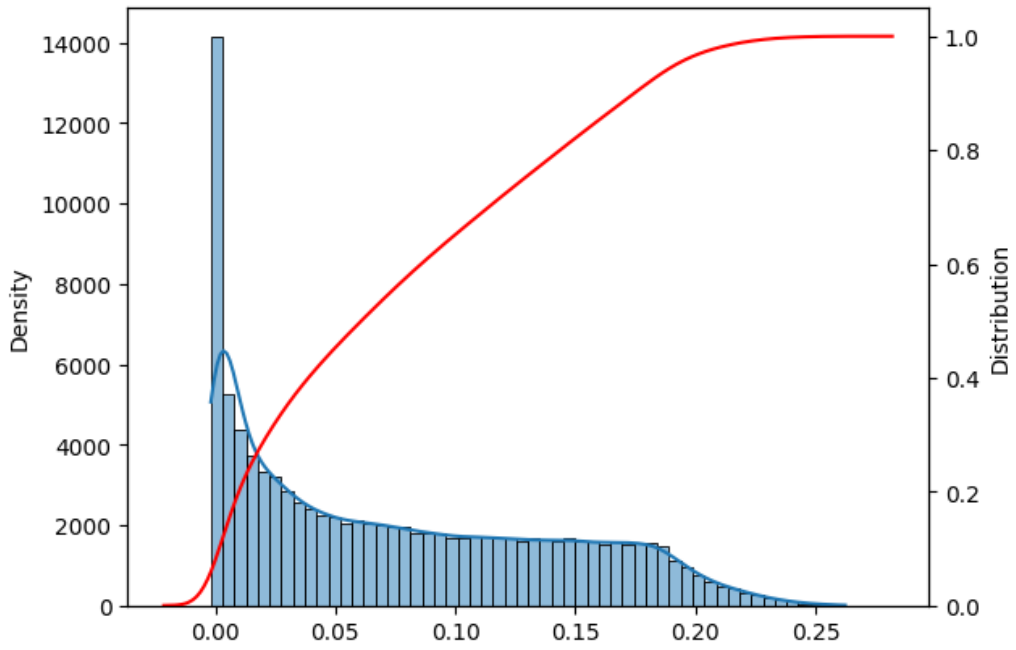
همانطور که در شکل۳ مشاهده می‌شود و با توجه به آنچه در فصل۳ گفته شد، با افزایش نرخ یادگیری میانگین خطا افزایش می‌یابد.

شکل ۴



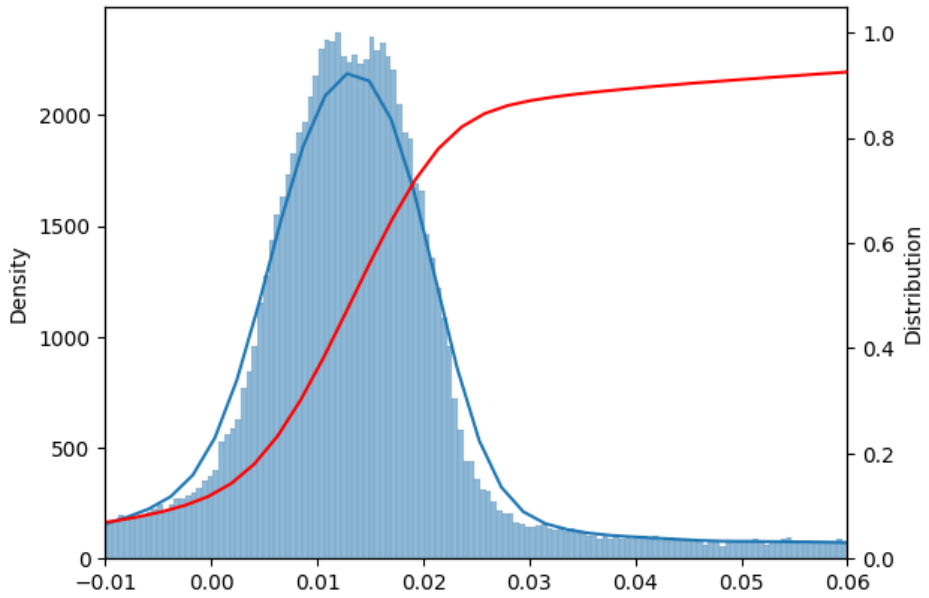
برای رسم این شکل داده ها را به ۸ قسمت تقسیم نموده ایم و در هر مرحله یک هشتم به آن افزودیم بدین صورت که در گام اول یک هشتم، در گام دوم دو هشتم، در گام سوم سه هشتم و... همانطور که ملاحظه می‌شود نتیجه با انتظار ما که فصل قبل به آن اشاره شد مطابقت داره یعنی یا افزایش داده میزان خطا کمتر و دقت افزایش می‌یابد.

شکل ۵



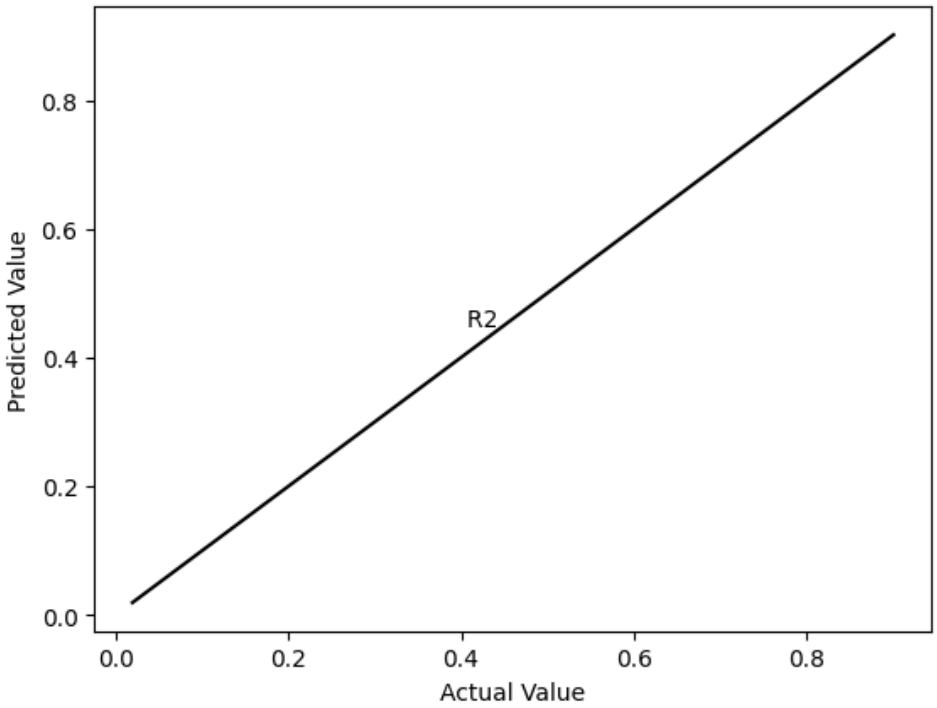
شکل۵ توزیع خطای قیمت آپشن واقعی و قیمت آپشن تخمین زده شده توسط شبکه را نشان می‌دهد.

شکل ۶



شکل۶ توزیع خطای نوسان ضمنی واقعی با نوسان ضمنی تخمین زده شده است که ملاحظه می‌شود از توزیع نرمال پیروی میکند.

شکل ۷



شکل۷ دقت R2 مقدار واقعی نوسان و مقدار تخمین زده شده نوسان است.

شبکه عصبی قیمت گذاری آپشن و نوسان ضمنی هستون

در این بخش نیز ابتدا باید مقادیر قیمت آپشن توسط مدل هستون با بسط کسینوسی محاسبه شود. ذکر این نکته ضروری است که بازه مقادیری که توسط مقاله گزارش شد با بازه بدست امده در این گزارش متفاوت بود لیکن ما مقادیر داده‌هایی که خارج از آن محدوده بود را حذف نمودیم (از یک میلیون داده استفاده نمودیم و تعداد داده‌هایی که حذف شد کمتر از ده درصد کل داده‌ها بود) .

جدول ۴

ANN Parameters	Range	Method
Input		
Moneyness, m = S0/K	(0.6, 1.4)	LHS
Time to maturity, τ	(0.1, 1.4)(year)	LHS
Risk free rate, r	(0.0%, 10%)	LHS

Correlation, ρ	(−0.95, 0.0)	LHS
Reversion speed, κ	(0.0, 2.0)	LHS
Long average variance, θ	(0.0, 0.5)	LHS
Volatility of volatility, σ	(0.0, 0.5)	LHS
Initial variance, ν0	(0.05, 0.5)	LHS
Output		
European call price, V	(0, 0.1)	

روش LHS که به عنوان نمونه برداری لاتین Hypercube نیز شناخته می شود، یک روش نمونه گیری آماری است که در زمینه های مختلف از جمله مهندسی، آزمایش های کامپیوتری و مطالعات شبیه سازی استفاده می شود. برای ایجاد یک نمونه نماینده از مقادیر پارامتر از یک توزیع چند متغیره طراحی شده است. در نمونه برداری لاتین Hypercube، فضای پارامتر به فواصل مساوی در طول هر بعد تقسیم می شود و سپس از هر بازه یک نمونه تصادفی انتخاب می شود. این تضمین می کند که نمونه به دست آمده کل فضای پارامتر را به طور یکنواخت و کارآمد پوشش می دهد. ساختار Hypercube لاتین به کاهش تنوع نمونه‌گیری کمک می‌کند و نمونه نماینده‌تری را در مقایسه با نمونه برداری تصادفی ساده تضمین می‌کند.

در قسمت قیمت گذاری آپشن مقادیر تخمین زده شده دقت بالا و خطای کمی داشت اما در قسمت محاسبه نوسان ضمنی به مشکل برخوردیم و دقت ۰.۰۵ رسیدیم لزا نوع شبکه گزارش شده در مقاله را عوض کردیم و درنهایت با انجام تغییرات زیاد سرانجام به دقت ۰.۶۹ رسیدیم. در حین عوض کردن شبکه متدهای مختلفی را امتحان کردیم مثل feature selection اماده برخلاف انتظار نتیجه خوبی نداشت.

- feature selectiob

انتخاب ویژگی فرآیند انتخاب زیر مجموعه ای از ویژگی های مرتبط (متغیرها، پیش بینی کننده ها) برای استفاده در ساخت یک مدل یادگیری ماشین است. هدف از انتخاب ویژگی بهبود عملکرد مدل با کاهش بیش از حد برازش، افزایش قابلیت تفسیر مدل و کاهش پیچیدگی محاسباتی است.
چندین روش برای انتخاب ویژگی وجود دارد، از جمله:

۱. روش های فیلتر: این روش ها از معیارهای آماری برای امتیاز دادن به هر ویژگی استفاده می کنند و بر اساس امتیاز، مرتبط ترین ها را انتخاب می کنند. نمونه‌هایی از روش‌های فیلتر عبارتند از انتخاب ویژگی مبتنی بر همبستگی، انتخاب ویژگی متقابل مبتنی بر اطلاعات، و انتخاب ویژگی مجذور کای.
۲. روش‌های Wrapper: این روش‌ها از یک الگوریتم یادگیری ماشینی خاص برای ارزیابی

عملکرد زیرمجموعه‌های مختلف ویژگی‌ها و انتخاب بهترین زیرمجموعه بر اساس ارزیابی استفاده می‌کنند. نمونه‌هایی از روش‌های wrapper شامل حذف ویژگی بازگشتی و انتخاب رو به جلو/عقب است.

۳. روش های Embedded: این روش ها انتخاب ویژگی را به عنوان بخشی از فرآیند آموزش مدل انجام می دهند. نمونه هایی از روش های تعبیه شده عبارتند ازLasso رگرسیون ، Ridge رگرسیون و روش های مبتنی بر درخت تصمیم مانند جنگل های تصادفی.

انتخاب روش انتخاب ویژگی به مشکل خاص و ویژگی های داده بستگی دارد. ارزیابی تأثیر انتخاب ویژگی بر عملکرد و قابلیت تفسیر مدل و جلوگیری از تطبیق بیش از حد با داده‌های آموزشی مهم است.

فصل ۴ نتایج و بحث

در مقاله یک رویکرد ANN برای کاهش زمان محاسبه گزینه‌های مالی قیمت‌گذاری، به ویژه برای مدل‌های مالی با ابعاد بالا پیشنهاد شد. نتایج عددی نشان می‌دهد که ANN می‌تواند قیمت آپشن‌ها و نوسانات ضمنی را به طور موثر و دقیق به روشی قوی محاسبه کند. این بدان معناست که، به ویژه برای فرآیندهای قیمت دارایی که منجر به محاسبات زمان‌برتر می‌شود، می‌توان یک تکنیک تقریب بسیار کارآمد را با استفاده از ANN ارائه کرد. اگرچه آموزش آفلاین در آن زمان بیشتر طول می کشد، اما پیش بینی آنلاین سریع خواهد بود. علاوه بر این، محاسبات موازی به حل‌کننده ANN اجازه می‌دهد تا قراردادهای مشتق را «در حالت دسته‌ای» پردازش کند و این ویژگی سرعت محاسباتی را تا حدود ۱۰۰ در GPUها نسبت به حل‌کننده اصلی در مورد فعلی افزایش می‌دهد. با توجه به مدل‌های دارایی با ابعاد بالا، تا زمانی که مقادیر گزینه را بتوان با هر حل‌کننده عددی (تکنیک فوریه، تفاضل‌های محدود یا روش مونت کارلو) بدست آورد، ممکن است با استفاده از یک ANN آموزش‌دیده، محاسبه را تسریع کنیم. اگرچه در این کار روی آپشن‌های خرید اروپایی شد، اما باید بتوان رویکرد قیمت‌گذاری گزینه‌های پیچیده‌تر مانند آمریکایی، برمودا یا گزینه‌های عجیب و غریب را گسترش داد. این کار در ابتدا امکان یادگیری یک حل کننده مبتنی بر داده را برای سرعت بخشیدن به حل مدل های مالی پارامتریک نشان می دهد. دقت مدل را می توان بیشتر بهبود بخشید، به عنوان مثال، با استفاده از شبکه های عصبی عمیق تر، معماری های NN پیچیده تر. سرعت حل کننده نیز ممکن است بهبود یابد، به عنوان مثال، با طراحی یک شبکه عصبی کم عمق تر.

منابع یا مراجع

- Liu, S., Oosterlee, C. W., & Bohte, S. M. (2019). Pricing Options and Computing Implied Volatilities using Neural Networks. Applied Sciences, 9(3), 486. doi: 10.3390/app9030486.

  https://arxiv.org/abs/1901.08943

- Fang, F., & Oosterlee, C. W. (2008). A novel pricing method for European options based on Fourier-cosine series expansions. SIAM Journal on Scientific Computing, 31(2), 826-848. doi: 10.1137/070679065

  https://faculty.baruch.cuny.edu/lwu/890/FanOosterlee2008.pdf

- The thesis is "The COS Method: An Efficient Fourier Method for Pricing Financial Derivatives." The author is Fang Fang, The thesis was defended on December 8, 2010, at the Delft University of Technology in the Netherlands.The address https://core.ac.uk/download/pdf/301638164.pdf#page=50&zoom=100,101,577.

- Dunn, R., Hauser, P., Seibold, T., & Gong, H. (2015). Estimating Option Prices with Heston's Stochastic Volatility Model. Department of Mathematics and Statistics, Valparaiso University. Retrieved from https://www.valpo.edu/mathematics-statistics/files/2015/07/Estimating-Option-Prices-with-Heston.

- S. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, Rev. Financ. Studies, 6 (1993), pp. 327–343.

  https://faculty.baruch.cuny.edu/lwu/890/Heston93.pdf

- Shinde, A. S., & Takale, K. C. (2012). Study of Black-Scholes Model and its Applications. Procedia Engineering, 38, 270-279.

  https://doi.org/10.1016/j.proeng.2012.06.035