بسمه تعالی

HEDGING WITH LINEAR REGRESSIONS AND NEURAL NETWORKS

(پروژه پایانی درس ریاضیات مالی)

فاطمه السادات موسوی، رعنا حسنی

بهار 1402

این گزارش شامل دو بخش اصلی است. در بخش 1 به تشریح مفاهیم مقاله از لحاظ تئوری پرداخته می‌شود. در این بخش ابتدا چکیده‌ای از هدف مقاله بیان شده و در ادامه مفاهیم اولیه مربوط به شبکه‌های عصبی مصنوعی و مدل‌های رگرسیون خطی که دو مفهوم اساسی مورد استفاده در این مقاله هستند، تشریح شده است.

در بخش 2 روند پیاده‌سازی کدهای هریک از بخش‌های شبیه‌سازی داده، شبکه‌های عصبی مصنوعی و مدل‌های رگرسیون خطی به ترتیب بیان شده و خروجی و مصورسازی‌های انجام شده آورده شده است.

بخش 1 – تشریح مقاله و مفاهیم تئوری

چکیده

این مقاله، استفاده از شبکه‌های عصبی مصنوعی (ANN) را برای مدیریت ریسک گزینه‌ها بررسی می‌کند و عملکرد آنها را با مدل‌های رگرسیون خطی ساده مقایسه می‌کند. هدف به حداقل رساندن واریانس خطای پوشش ریسک در معاملات روزانه اختیارات است. این مطالعه یک مدل یک دوره‌ای را در نظر می‌گیرد که در آن اپراتور می‌تواند برای کاهش واریانس پورتفولیو، کالای اساسی را بخرد یا بفروشد.

هدف این مقاله تعیین نسبت پوشش (Δ) برای به حداقل رساندن واریانس پرتفوی پوشش‌دهی شده است. هر دو مدل ANN و رگرسیون خطی بر روی مجموعه داده های درون نمونه (IN-SAMPLE) آموزش داده شده و بر روی مجموعه داده خارج از نمونه (OUT-SAMPLE) ارزیابی می شوند و واریانس پرتفوی پوشش داده شده با میانگین مربعات خطای پوششی (MSHE) تقریب زده می‌شود.

ارزش پرتفوی قابل پوشش ریسک (VΔ1) برای روز معاملاتی بعدی به صورت زیر ارائه می‌شود:

$$V\Delta 1 = \Delta S1 + (1 + RONR\Delta T)\,(C0 - \Delta S0) - C1$$

- Δ نسبت پوششی است که نشان دهنده تعداد سهام دارایی پایه (S) برای خرید یا فروش است.
- S0و S1 قیمت دارایی های اساسی در ابتدا و انتهای دوره پوشش هستند.
- C0 و C1 قیمت های اختیار خرید یا فروش در ابتدا و انتهای دوره پوشش هستند.
- RONR نرخ یک شبه است که اپراتور می تواند با آن پول قرض کند یا وام دهد.
- Δ T طول دوره پوشش، معمولا یک روز است.

هدف اپراتور به حداقل رساندن میانگین مربعات خطای پوششی (MSHE) است که می تواند به صورت تقریبی باشد:

$$\mathrm{var}(V_1^{\delta}) \approx \mathrm{MSHE} = \frac{1}{N_{test}} \sum_{t,j}^{N_{test}} \left(100 \frac{V_{t+1,j}^{\delta}}{S_t} \right)^2,$$

- NTEST تعداد نمونه ها در مجموعه داده خارج از نمونه است.
- VΔT+1, J ارزش پرتفوی پوشش‌دهی‌شده برای گزینه J در پایان دوره پوشش در روز معاملاتی (T+1) است.
- ST قیمت دارایی پایه در پایان روز معاملاتی T است.

شبکه‌های عصبی مصنوعی (ANN) – توضیح مفاهیم اولیه

پیش از توضیح کد مربوط به پیاده‌سازی شبکه‌های عصبی، بهتر است مفاهیم پایه‌ای از شبکه‌های عصبی را به اختصار توضیح دهم:

- شبکه‌های عصبی (NEURAL NETWORKS)

شبکه‌های عصبی مدل‌های ریاضی هستند که از ساختار مغز انسان الهام گرفته‌اند و برای حل مسائل پیچیده مورد استفاده قرار می‌گیرند. این مدل‌ها از چندین لایه از واحدهای محاسباتی (نورون‌ها) تشکیل شده‌اند و می‌توانند الگوها و ارتباط‌های پیچیده را از داده‌ها یاد بگیرند.

- نورون‌ها (NEURONS)

نورون‌ها واحدهای پایه‌ای شبکه‌های عصبی هستند. هر نورون ورودی‌های خود را با وزن‌ها ضرب می‌کند، مقدار آن‌ها را جمع می‌کند و نتیجه را با استفاده از تابع فعال‌سازی محاسبه می‌کند.

- تابع فعال‌سازی (ACTIVATION FUNCTION)

تابع فعال‌سازی نقش بسیار مهمی در عملکرد شبکه‌های عصبی ایفا می‌کند. این تابع روی خروجی نورون اعمال می‌شود و امکان غیرخطی بودن مدل را فراهم می‌کند. از توابع فعال‌سازی معروف می‌توان به RELU، SIGMOID و TANH اشاره کرد.

- تابع هزینه (LOSS FUNCTION)

تابع هزینه نشان می‌دهد میزان اختلاف بین خروجی مدل و مقدار واقعی داده‌ها چقدر است. هدف شبکه عصبی این است که با کمینه کردن مقدار تابع هزینه، پارامترهای خود را به گونه‌ای تنظیم کند که خروجی مدل بهترین عملکرد را داشته باشد. معروف‌ترین تابع هزینه معمولی است که در شبکه‌های عصبی استفاده می‌شود، MEAN SQUARED ERROR (MSE) یا میانگین مربعات خطا که در این مقاله از همین مقدار برای ارزش سبد پورتفوی پوشش داده شده استفاده می‌شود. (HMSE)

- رگولاریزیشن (REGULARIZATION)

رگولاریزیشن یک تکنیک استفاده می‌شود تا از بیش‌برازش (OVERFITTING) در مدل جلوگیری کند. این تکنیک با اضافه کردن جملاتی به تابع هزینه موجب کاهش مقادیر وزن‌ها می‌شود و در نتیجه مدل عمومی‌تر و قابلیت تعمیم بیشتری به داده‌های جدید دارد.

- نحوه‌ی کارکرد شبکه‌های عصبی و الگوریتم BACK PROPAGATION

یکی از روش‌های رایج آموزش شبکه‌های عصبی، الگوریتم BACKPROPAGATION است. این روش الگوریتم (GRADIENT DESCENT) را برای به‌روزرسانی وزن‌ها به‌کار می‌گیرد. این فرآیند به این صورت انجام می‌شود:

1. در ابتدا، وزن‌های شبکه به‌صورت تصادفی مقداردهی می‌شوند.

2. برای هر داده آموزشی، فرآیند FEEDFORWARD انجام می‌شود که شبکه خروجی مدل را برای ورودی‌ها محاسبه می‌کند.

3. با مقایسه خروجی مدل با مقدار واقعی داده‌ها، خطای مدل محاسبه می‌شود (براساس تابع هزینه).

4. سپس از الگوریتم BACKPROPAGATION برای محاسبه گرادیان تابع هزینه نسبت به وزن‌ها استفاده می‌شود.

5. با استفاده از گرادیان‌ها، وزن‌ها به‌روزرسانی می‌شوند تا خطا کمینه شود.

6. این فرآیند تکرار می‌شود تا وزن‌ها به مقادیری برسند که مدل عملکرد مناسبی داشته باشد.

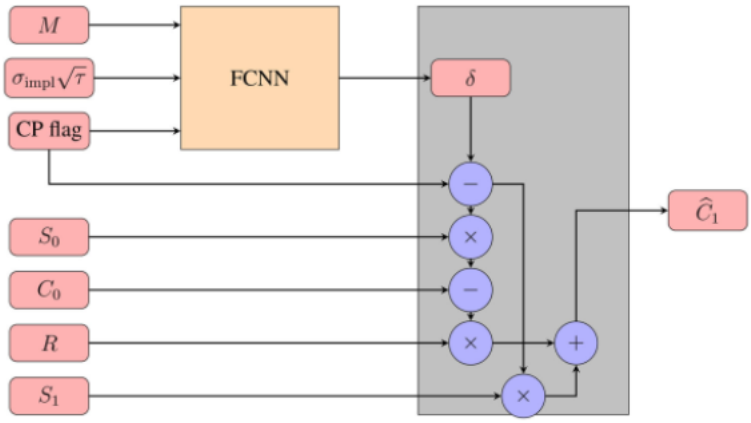
- انتخاب بهترین هایپرپارامتر و آموزش مدل

با ورودی دادن داده‌های آموزش و سنجش عملکرد مدل روی داده‌های اعتبارسنجی می‌توان معیاری از این که مدل چقدر خوب است به دست آورد. از این روش برای انتخاب بهترین هایپرپارامتر برای مدل استفاده می‌شود. بدین ترتیب که مدل‌های گوناگونی را با استفاده از هایپرپارامترهای مختلف ساخته و عملکرد آن را روی داده‌های اعتبارسنجی ارزیابی کرده و بدین ترتیب بهترین مدل را برای روند اصلی آموزش و ارزیابی روی داده‌های تست می‌گزینیم.

- ارزیابی عملکرد مدل

عملکرد مدل بر روی داده‌های اعتبارسنجی و تست با معیار هزینه (تابع هزینه) اندازه‌گیری می‌شود. هدف شبکه عصبی این است که معیار هزینه را به حداقل برساند. بدین ترتیب HMSE نیز مینیمم می‌شود. این بدین معناست که پیش‌بینی‌های مدل دقیق‌تر شده و خطاها کاهش می‌یابند.

ساختار شبکه عصبی استفاده شده در این مقاله به صورت زیر است:



در شمای بالا M = S0/K و ΣIMPL√T پارامترهای مالی هستند که به ترتیب نشان‌دهندهٔ (MONEYNESS) و جذر واریانس تضمین شده کلی (TOTAL IMPLIED VARIANCE) هستند. "CP FLAG" پرچمی بولین برای تعیین نوع گزینه (CALL یا PUT) است. S0 و S1 قیمت‌های دارایی مبنا در ابتدا و پایان دوره هستند، C0 قیمت گزینه در ابتداي دوره را نشان می‌دهد و C1 مقدار تکثیر را نمایش می‌دهد. در نهایت، R = 1 + RONRT بازده بدون ریسک در طول شبانه روز است.

همانطور که در شمای نمایش داده شده نیز مشاهده می‌شود، فرمول‌های زیر پیاده‌سازی شده‌اند:

Δ = FCNN(M, ΣIMPL√T)

VΔ1 = ΔS1 + (1 + RONRT)(C0 - ΔS0) - C1 => C1^HAT = ΔS1 + (1 + RONRT)(C0 - ΔS0)

در این مقاله محاسبه یا برآورد قیمت گزینه و سپس استفاده از حساسیت‌های آن به عنوان نسبت‌های HEDGING هدف نیست و به جای آن نسبت هجینگ، یعنی کمیت مورد نظر ما، به صورت مستقیم در شبکه عصبی تخمین زده می‌شود. به همین دلیل، در مقاله یک ANN به نام HEDGENET طراحی شده است که دارای دو قسمت است.

بخش اول، یک شبکه عصبی چندلایه و کاملاً متصل (FULLY CONNECTED)(FCNN) است و از ویژگی‌ها یک نسبت HEDGING پیشبینی می‌کند که سپس بوسیلهٔ بخش دوم به مقدار C1^HAT = V1 + C1 تبدیل می‌شود. این خروجی HEDGENET سپس با استفاده از کاهش مجموع مربع اختلافات با قیمت‌های مشاهده شدهٔ C1 در پایان هر دوره، آموزش داده می‌شود.

شبکه عصبی مورد استفاده در این مقاله دارای دو لایه با 30 نود در هر لایه است.

شبکه‌ی عصبی با استفاده از مجموعه ویژگی‌های مختلفی ساخته و آموزش داده شده است. این مجموعه ویژگی‌ها به طور دقیق‌تر در بخش پیاده‌سازی تشریح می‌شوند.

مدل‌های رگرسیون خطی – توضیح مفاهیم اولیه

در این مطالعه، محققان استفاده از مدل‌های آماری و رگرسیون خطی را برای قیمت‌گذاری گزینه‌ها و استراتژی‌های پوشش ریسک بررسی می‌کنند. هدف آنها بهبود عملکرد پوشش با در نظر گرفتن سایر حساسیت ها علاوه بر دلتای بلک شولز (Δ_BS) است.

در ابتدا مفاهیم استفاده شده در مقاله را توضیح می‌دهیم:

در معاملات اختیار معامله و مدیریت ریسک، "GREEKS" به مجموعه ای از حساسیت های کلیدی اشاره می کنند که چگونگی تغییر قیمت یک اختیار معامله را در پاسخ به عوامل مختلف، مانند تغییرات در قیمت دارایی پایه، زمان تا انقضا، و نوسانات ضمنی توضیح می دهد. GREEKS نقش مهمی در درک و مدیریت ریسک مرتبط با موقعیت‌های گزینه دارد. در این مطالعه از GREEK های زیر استفاده شده است:

دلتا:(Δ)

دلتا حساسیت قیمت یک اختیار معامله را نسبت به تغییرات قیمت دارایی پایه اندازه گیری می کند. این نشان دهنده تغییر مورد انتظار در قیمت اختیار معامله برای تغییر یک واحدی در قیمت دارایی پایه است. دلتا از -1 تا 0 برای گزینه های PUT و 0 تا 1 برای گزینه های CALL متغیر است.

فرمول دلتا:

برای گزینه های CALL : $\Delta = \partial C / \partial S$

برای گزینه های PUT : $\Delta = \partial P / \partial S$

- C = قیمت گزینه تماس

- P =قیمت اختیار خرید
- S =قیمت دارایی پایه

وگا:(V)

VEGAحساسیت قیمت یک گزینه را به تغییرات در نوسانات ضمنی اندازه گیری می کند. این نشان دهنده تغییر مورد انتظار در قیمت اختیار معامله برای تغییر یک درصدی در نوسانات ضمنی است.

فرمول وگا:

برای هر دو گزینه PUT و CALL : $V = \partial C / \partial \sigma$

- C = قیمت گزینه CALL یا PUT
- σ = نوسانات ضمنی قیمت گزینه

گاما:(Γ)

گاما نرخ تغییر دلتای یک اختیار را با توجه به تغییرات در قیمت دارایی پایه اندازه گیری می کند. این توضیح می دهد که چگونه دلتا با تغییر قیمت دارایی اساسی تغییر می‌کند.

فرمول گاما:

برای هر دو گزینه PUT و CALL : $\Gamma = \partial^2 C / \partial S^2$

- C = قیمت گزینه CALL یا PUT
- S = قیمت دارایی پایه

وانا:(VA)

VANNA نرخ تغییر دلتای یک گزینه را با توجه به تغییرات در نوسانات ضمنی اندازه گیری می کند. این حساسیت نشان می دهد که چگونه دلتا با تغییر نوسانات ضمنی تغییر می‌کند.

فرمول برای وانا:

برای هر دو گزینه PUT و CALL :

$VA = \partial^2 C / \partial S \partial \sigma$

- C = قیمت گزینه CALL یا PUT
- S = قیمت دارایی پایه
- σ =نوسانات ضمنی قیمت گزینه

ولاتیلیتی ضمنی (IMPLIED VOLITILITY):

ولاتیلیتی ضمنی (IMPLIED VOLITILITY) همبستگی منفی بین بازده مشاهده شده و نوسانات آن‌ها را در بازارهای سهام توصیف می کند. این اثر بدان معناست که وقتی قیمت سهام کاهش می‌یابد، نوسانات آن افزایش می‌یابد و بالعکس. این اثر در مطالعات مختلف مشاهده و تایید شده است. پس در این مقاله، در مدل‌های رگرسیون خطی پیشنهادی برای پوشش، از نوسانات ضمنی (V_BS) به عنوان یکی از متغیرهای مستقل علاوه بر دلتای بلک شولز (Δ_BS) استفاده می‌شود.

مدل رگرسیون خطی:

مدل رگرسیون خطی شامل مقادیر Δ_BS ، V_BS، VANNA (VA_BS)، و گاما (Γ_BS) است.

پارامترهای(A,B,C,D) به طور جداگانه برای PUT و CALL تخمین زده می‌شوند.

$\Delta\_LR = A^* \Delta\_BS + B^* V\_BS + C^* VA\_BS + D^* \Gamma\_BS.$

گنجاندن حساسیت‌های V_BS و Γ_BS اجازه می‌دهد تا نسبت پوشش ریسک را بر اساس پولی بودن و زمان سررسید گزینه تنظیم کنید که منجر به بهبود عملکرد پوشش ریسک می‌شود.

مدل HULL-WHITE

نسبت پوشش پیشنهادی (Δ_HW) شامل Δ_BS وV_BS ، با در نظر گرفتن زمان تا سررسید (T)و سایر پارامترها (A,B,C,D) است.

$$\Delta\_HW = \Delta\_BS + [V\_BS/(\sqrt{T} * S)] * (A + B * \Delta\_BS + C * \Delta\_BS^2)$$

هال و وایت (2017) این مدل را از تجزیه و تحلیل دقیق گزینه های S&P 500 به دست آوردند که عملکرد پوشش دهی عالی را در شاخص های مختلف نشان می‌داد. در نتیجه، این مطالعه نشان می‌دهد که ترکیب حساسیت‌های اضافی، مانند V_BS و Γ_BS، در مدل‌های آماری و رگرسیون خطی می‌تواند استراتژی‌های پوشش ریسک فراتر از رویکرد سنتی دلتای BLACK-SCHOLES را افزایش دهد و منجر به بهبود مدیریت ریسک در بازارهای مالی شود.

بخش 2 – پیاده‌سازی و نتایج و خروجی‌ها

1. شبیه‌سازی داده‌ها

شبیه‌سازی داده‌ها برای پوشش ریسک با روش دلتای بلک شولز شامل چندین گام است. در اینجا یک توضیح گام به گام را که همراه با فرمول‌های استفاده شده است، به اختصار توضیح می‌دهیم:

- گام 1

افق زمانی را تعریف کرده و آن را گسسته می‌کنیم: مدت زمانی را که می خواهیم داده ها را شبیه سازی کنیم، مشخص می‌کنیم و این بازه زمانی را به فواصل یا مراحل زمانی کوچکتر (ΔT) تقسیم می‌کنیم.

- گام 2

نرخ بهره را تا انقضای مشخص شده با درون‌یابی محاسبه می کنیم.

- گام 3

در این گام اطلاعات مربوط به فردا در یک DATAFRAME را ترکیب می‌کنیم و نقاط داده آتی مانند سهام فردا، حجم، یا سایر اطلاعات مرتبط را در یک چارچوب داده می‌گنجانیم.

- گام 4

محاسبه قیمت OPTION ها با استفاده از فرمول بلک-شولز: برای هر قیمت دارایی شبیه سازی شده، قیمت اختیار مربوطه را با استفاده از فرمول BLACK-SCHOLES محاسبه می‌کنیم:

CALL OPTION PRICE = ST * N(D1) - K * EXP(-R * T) * N(D2)

PUT OPTION PRICE = K * EXP(-R * T) * N(-D2) - ST * N(-D1)

- ST قیمت دارایی شبیه سازی شده
- K قیمت عملیاتی یا قیمتی که برای پایان سررسید مشخص شده است.
- R نرخ بهره بدون ریسک
- T زمان انقضا
- N () تابع توزیع تجمعی یک توزیع نرمال استاندارد
- D1 و D2 که به شکل زیر قابل محاسبه هستند:

D1 = (LN(ST / K) + (R + σ ^2/2) * T) / (σ * $\sqrt{}$ T)

D2 = D1 - σ * $\sqrt{}$ T

- گام 5

دلتاهای OPTION را محاسبه می‌کنیم: برای هر قیمت دارایی شبیه سازی شده، دلتای گزینه مربوطه را با استفاده از فرمول BLACK-SCHOLES محاسبه می کنیم:

CALL OPTION DELTA = N(D1)

PUT OPTION DELTA = N(D1) - 1

- N() تابع توزیع تجمعی یک توزیع نرمال استاندارد
- D1 همانطور که در گام 4 محاسبه شد

- گام 6

GREEKها (حساسیت‌های آپشن) شامل به ترتیب تتا ، گاما ، وگا ، وانا محاسبه و در DATAFRAME ذخیره می کنیم.

$$N'(d_1) = \frac{e^{\frac{-d_1^2}{2}}}{\sqrt{2\pi}} \qquad \theta_{call} = -\frac{S_0 N'(d_1)\sigma}{2\sqrt{T)} - r.\, Ke^{-rT}\, N\,(d_2)}$$

$$\Gamma = \frac{N'(d_1)}{S\sigma\sqrt{T}}$$

$$\nu = S_0\sqrt{T}\, N'\,(d_1)$$

$$vanna = \sqrt{T-t}\, N'(d_1)(\frac{d_2}{\sigma})$$

- گام 7

در اینجا از روش برنت استفاده می‌کنیم تا به طور مکرر نوسانات ضمنی را که با قیمت گزینه داده شده مطابقت دارد، بیابد و آنها را برای هر OPTION محاسبه کنیم.

- گام 8

نرمال‌سازی ستونی را بر روی ستون‌های مشخص شده در DATAFRAME انجام می‌دهیم.

- گام 9

در این بخش یک کلاس را تعریف می‌کنیم که نشان دهنده یک گزینه‌ی اروپایی با ویژگی‌های خاص مانند قیمت اعتصاب، تاریخ شروع و تاریخ سررسید است و قیمت‌های اساسی گزینه را در محدوده زمانی مشخص ذخیره می‌کنیم.

- گام 10

موقعیت های پوشش(HEDGE) را محاسبه می‌کنیم: تعداد OPTION های مورد نیاز برای پوشش در برابر تغییرات قیمت دارایی پایه را با ضرب دلتای CALL در تعداد سهام یا قراردادهای ارائه شده توسط هر اختیار تعیین می‌کنیم. موقعیت های پوشش را در هر گام زمانی بر اساس تغییرات قیمت دارایی های شبیه سازی شده تنظیم می‌کنیم.

- گام 11

نرمال‌سازی ستونی را بر روی ستون‌های مشخص شده در DATAFRAME انجام می دهیم.

- گام 12

به منظور تولید OPTION ها برای یک مسیر قیمت معین بر اساس قوانین بورس گزینه‌های هیئت مدیره شیکاگو (CBOE) به صورت زیر عمل می‌کنیم و یک بازه با تعداد گام مشخص شده و محدودیت خاص از روی قیمت سهام برای قیمت STRICK می‌سازیم. اگر قیمت لحظه ای سهام از این بازه خارج شد یک STRICK جدید می‌سازیم و برای هرکدام سررسید را محاسبه کرده و برای هر سررسید و بازه قیمت STRICK , OPTION ایجاد می‌کنیم . سپس یک سری موارد مانند خارج نشدن قیمت لحظه ای از بازه ی STRICK برای هر سررسید و منقضی نشدن OPTION ها تا تاریخ سررسیدشان و... را بررسی می‌کنیم. در انتها بعد از بررسی‌ها یک OPTION جدید می‌سازیم.

- گام 13

دارایی پایه‌ای S0 را با استفاده از حرکت براونی شبیه سازی کرده و در ستون"S0" ذخیره می کنیم:

ST = S0 * EXP((M - σ ^2/2) * T + σ * $\sqrt{}$ (Δ T) * ZT)

- ST قیمت شبیه سازی شده در زمان T برای ستون S0
- S0 قیمت اولیه
- M بازده مورد انتظار
- σ نوسان
- ΔT گام زمانی
- ZT یک عدد تصادفی

- گام 14

یک DATAFRAME را برمی‌گردانیم که حاوی اطلاعات مختلفی در مورد گزینه ها، مانند قیمت سهام، قیمت اعتصاب، زمان تا سررسید و ... است.

- گام 15

نمودار برای شبیه سازی قیمت سهام مسیر درون نمونه(IN-SAMPLE) را به همراه 10 مسیر خارج از نمونه (OUT-SAMPLE) برای داده‌های شبیه سازی داده شده ترسیم می‌کنیم که تکامل قیمت سهام را در طول زمان نشان می‌دهد و دوره های آموزشی(TRAIN)، اعتبارسنجی(VALIDATION) و آزمون(TEST) را به صورت بصری متمایز می‌کند که محور X «تاریخ»، محور Y «قیمت سهام» است. این نمودار در زیر آورده شده است:

- گام 16

در اینجا یک مجموعه داده تمیز از OPTION ها را بر اساس یک مسیر قیمت سهام شبیه سازی شده، با استفاده از پیش پردازش داده‌ها و محاسبه معیارهای مربوط به اختیار برای داده های نرمال سازی شده ، تولید می‌کنیم.

- گام 17

این گام، با فراخوانی داده های تمیز را با پارامترهای مشخص، یک مجموعه داده TRAIN و VALIDATION ایجاد می‌کند و مجموعه داده ترکیبی را به عنوان یک فایل CSV ذخیره می‌کنیم.

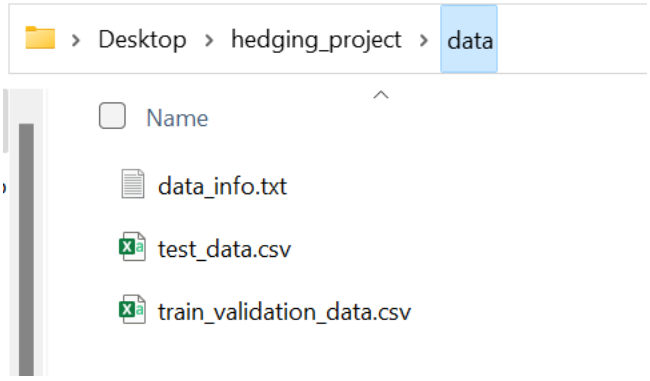
- گام 18

در این گام با تنظیم پارامترها، ترسیم قیمت سهام و به روز رسانی پارامترهای لازم برای مجموعه‌های TRAIN، مجموعه های تست مونت کارلو را تولید می‌کنیم. در نهایت، یک دایرکتوری برای ذخیره مجموعه داده‌های آزمایشی مونت کارلو ایجاد کرده و با فراخوانی تابع مربوطه نمودار گام 16 را رسم می‌کنیم.

- گام 19

فایلی را که شامل OPTION ها با پارامترهای محاسبه شده است، ذخیره می‌کنیم.

در پایان دایرکتوری فایل‌های حاصل از شبیه‌سازی به شرح زیر است:



فایل DATA_INFO.TXT حاوی اطلاعاتی از داده‌های تولید شده و فایل‌های TEST_DATA.CSVو TRAIN_VALIDATION_DATA.CSV به ترتیب حاوی داده‌های تست و داده های آموزش و اعتبار سنجی هستند.

شمایی از فایل نهایی داده‌ها:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		date	S0	K	tau0	optionid	short_rate	M0	r	implvol0	V0	cp_int	S_1D	V_1D	implvol_1D
2	0	########	2791.26	2295	0.071146	1	0	1.216235	0	0.2	496.2638	0	2847.743	552.7431	0.2
3	1	########	2847.743	2295	0.067194	1	0	1.240846	0	0.2	552.7431	0	2876.169	581.1695	0.2
4	2	########	2876.169	2295	0.063241	1	0	1.253233	0	0.2	581.1695	0	2812.082	517.0825	0.2
5	3	########	2812.082	2295	0.059289	1	0	1.225308	0	0.2	517.0825	0	2896.885	601.8853	0.2
6	4	########	2896.885	2295	0.055336	1	0	1.262259	0	0.2	601.8853	0	2831.035	536.0346	0.2
7	5	########	2831.035	2295	0.051383	1	0	1.233566	0	0.2	536.0346	0	2912.438	617.438	0.2
8	6	########	2912.438	2295	0.047431	1	0	1.269036	0	0.2	617.438	0	2907.642	612.6415	0.2
9	7	########	2907.642	2295	0.043478	1	0	1.266946	0	0.2	612.6415	0	2960.23	665.2304	0.2
10	8	########	2960.23	2295	0.039526	1	0	1.289861	0	0.2	665.2304	0	2916.052	621.0518	0.2
11	9	########	2916.052	2295	0.035573	1	0	1.270611	0	0.2	621.0518	0	2911.671	616.6708	0.2
12	10	########	2911.671	2295	0.031621	1	0	1.268702	0	0.2	616.6708	0	2894.468	599.4684	0.2
13	11	########	2894.468	2295	0.027668	1	0	1.261206	0	0.2	599.4684	0	2908.049	613.0492	0.2
14	12	########	2908.049	2295	0.023715	1	0	1.267124	0	0.2	613.0492	0	2920.886	625.8858	0.2
15	13	########	2920.886	2295	0.019763	1	0	1.272717	0	0.2	625.8858	0	2910.005	615.0047	0.2
16	14	########	2910.005	2295	0.01581	1	0	1.267976	0	0.2	615.0047	0	2859.729	564.7292	0.2
17	15	########	2859.729	2295	0.011858	1	0	1.246069	0	0.2	564.7292	0	2847.572	552.5716	0.2
18	16	########	2847.572	2295	0.007905	1	0	1.240772	0	0.2	552.5716	0	2810.113	515.1134	0.2

2. پیاده‌سازی و نتایج شبکه‌های عصبی

در ادامه به توضیح دقیق‌تری از نحوه‌ی پیاده‌سازی شبکه‌ی عصبی در این پروژه می‌پردازیم. بدین منظور یک ساختار کلی برای شبکه‌ی عصبی در نظر می‌گیریم. این معماری با استفاده از کلاس HEDGENET پیاده‌سازی شده است. در پیاده‌سازی‌های این بخش از کتابخانه‌ی KERAS در PYTHON استفاده شده است. برای تشریح بهتر این بخش، مراحل پیاده‌سازی را به چند گام تقسیم‌بندی کرده‌ایم. در ادامه به توضیح هر یک از این گام‌ها می‌پردازیم.

- گام 1

تعریف معماری شبکه:

ابتدا معماری شبکه عصبی برای HEDGHING تعریف می‌شود. این معماری شامل لایه‌های

ورودی برای ویژگی‌ها و پارامترهای کنترلی مورد استفاده در مدل است. همچنین لایه‌های پنهان با تعداد نودها و توابع فعال‌سازی مختلف نیز تعریف می‌شوند و لایه خروجی برای محاسبه مقادیر مورد نیاز نیز مشخص می‌شود.

معماری مدل شبکه عصبی HEDGENET به شرح زیر است:

- مقادیر ورودی:

1. لایه 'FEATURES': این لایه ورودی برای ویژگی‌های استفاده شده در مدل است.
2. 'CP_INT': این ورودی برای پارامتر کنترلی است که نشان‌دهنده خرید (CALL) یا فروش (PUT) (۰ یا ۱) است.
3. 'V0': این ورودی برای ارزش اولیه نمونه‌های نمودارها است.
4. 'S0': این ورودی برای قیمت اولیه سهام است.
5. 'S1': این ورودی برای قیمت نهایی سهام است.

6. 'ON_RET': این ورودی برای بازده یک روزه سهام است.

- لایه‌های مخفی:

این مدل دارای چندین لایه مخفی است، که تعداد نورون‌ها توسط پارامتر 'NODES_PER_LAYER' هنگام ساخت مدل مشخص می‌شود. هر لایه مخفی از تابع فعال‌سازی RELU و تنظیم مدل (L2 REGULARIZATION) برای جلوگیری از بیش برازش استفاده می‌کند.

- لایه خروجی:

'OUT_TRAINABLE': این لایه خروجی نماینده مقادیر دلتای پیش‌بینی‌شده قبل از اعمال پارامتر کنترلی مربوط به CALL یا PUT است.

- محاسبات نهایی:

مقادیر دلتای پیش‌بینی‌شده منهای ورودی 'CP_INT' می‌شوند تا مقادیر نهایی دلتا ('DELTA' LAYER) به‌دست آید.

سپس از مقادیر 'DELTA' برای محاسبه ارزش نهایی پورتفوی ('V1_HAT') استفاده می‌شود.

- تنظیمات مدل:

مدل با بهره‌گیری از بهینه‌ساز آدام (ADAM OPTIMIZER) با نرخ یادگیری ('LR') مشخص‌شده و تابع خطا میانگین مربعات خطای واریانس پورتفوی HEDGE شده ('HMSE') برای آن در نظر گرفته می‌شود.

- توابع کلاس HEDGENET:

کلاس HEDGENET شامل متدهایی مانند BUILD_MODEL برای ساخت مدل، FIT برای آموزش مدل و CALCULATE_DELTA برای محاسبه مقادیر دلتا است.

در اینجا مختصراً به توضیح هر متد پرداخته می‌شود:

1. __INIT__(SELF)

این متد سازنده کلاس است که متغیر `SELF.MODEL` را مقداردهی اولیه می‌کند و به صورت پیش‌فرض مقدار آن را NONE قرار می‌دهد.

2. BUILD_MODEL

این متد مدل شبکه عصبی را بر اساس ویژگی‌های ورودی ساختاردهی می‌کند. ورودی‌های این متد شامل ابعاد ویژگی‌ها (`FEATURE_SHAPE`)، تعداد نودها در هر لایه مخفی (`NODES_PER_LAYER`)، نرخ یادگیری برای بهینه‌سازی (`LR`)، تابع فعال‌سازی خروجی (`OUTACT`)، تابع هزینه (`LOSS`) و معیارهای ارزیابی (`METRICS`) می‌شود.

3. FIT

این متد مدل را با داده‌های آموزش و اعتبارسنجی آموزش می‌دهد و عملکرد مدل را بر روی داده‌های اعتبارسنجی ارزیابی می‌کند. ورودی‌های این متد شامل داده‌های آموزش (`TRAIN_DATA`) و ارزیابی (`VAL_DATA`)، ویژگی‌های استفاده‌شده برای آموزش (`USED_FEATURES`)، نام ستون مربوط به خروجی مورد نظر (V1) همان مقدار پورتفوی نهایی، تعداد تکرارها (EPOCHS)، اندازه دسته‌ها (BATCH_SIZE) است.

4. CALCULATE_DELTA

این متد با استفاده از مدل آموزش داده‌شده و داده‌های ورودی، مقادیر دلتا را پیش‌بینی می‌کند. ورودی‌های این متد شامل داده‌های ورودی (DF) DATAFRAME و ویژگی‌های مورد استفاده برای پیش‌بینی (USED_FEATURES) می‌شود.

- گام 2

تعریف توابع کمکی برای آماده‌سازی داده‌ها و بارگذاری و پاکسازی داده‌های آموزش، اعتبارسنجی و تست:

در گام دوم توابعی پیاده‌سازی می‌شوند که برای پیش‌پردازش داده‌ها استفاده می‌شود تا داده‌های آموزش و اعتبارسنجی را از یکدیگر جدا کند و سپس ویژگی‌های مورد نیاز را استانداردسازی کند. سپس، داده‌های تست نیز برای ارزیابی و تحلیل آماده می‌شود.

توابع کمکی پیاده‌سازی شده در این بخش به شرح زیرند:

1. STANDARDIZE_FEATURE:

این تابع برای استانداردسازی ویژگی‌های انتخاب شده در دیتافریم یا لیستی از دیتافریم‌ها با استفاده از SCALER مشخص شده به کار می‌رود. ویژگی‌های استانداردسازی شده با نام‌های جدیدی که دارای پسوند `T_` هستند نمایش داده می‌شوند. خروجی این تابع دیتافریم‌های استانداردسازی شده است.

2. CALCULATE_PNL

این تابع برای محاسبه سود یا زیان (PNL) بر اساس مقادیر دلتای پیش‌بینی شده به کار می‌رود. مقادیر PNL محاسبه شده نشان‌دهنده سود یا زیانی است که بر اساس پیش‌بینی‌ها نسبت به مقادیر واقعی به دست می‌آید.

3. REMOVE_COLUMNS_RENAME

این تابع برای حذف ستون‌های مشخصی از یک دیتافریم یا چند دیتافریم و نیز تغییر نام ستون‌ها بر اساس دیکشنری‌های ارائه شده به کار می‌رود. این امکان را فراهم می‌کند که به صورت انتخابی ستون‌ها را از دیتافریم حذف و نام‌های خاصی را برای ستون‌ها تغییر دهیم.

4. ASSIGN_DATA_TAG

این تابع برای اختصاص برچسب‌های داده به یک دوره مشخص در دیتافریم ورودی استفاده می‌شود. برچسب داده شده نشان‌دهنده این است که داده‌ها به مجموعه آموزش (TRAIN)، اعتبارسنجی (VALIDATION) یا آزمون (TEST) تعلق دارند.

5. ADD_CUSTOM_FEATURES

این تابع ویژگی‌های سفارشی به دیتافریم ورودی اضافه می‌کند. این ویژگی‌ها بر اساس محاسباتی بر روی ستون‌های موجود در دیتافریم ایجاد می‌شوند. برای مثال، ویژگی اختصاصی 1 به صورت جذر مربعی از ستون TAU0 ضرب‌در ستون IMPLVOL0 محاسبه می‌شود. این ویژگی‌ها همان ویژگی‌های مطلوب ذکر شده در مقاله برای آموزش شبکه‌ی عصبی هستند.

6. MODIFY_DATAFRAME

این تابع از دیتافریم ورودی، نمونه‌های IN-THE-MONEY و همچنین داده‌هایی که در محدوده مقداری مورد نظر MONEYNESS نیستند، حذف می‌کند. سپس تعداد نمونه‌های حذف‌شده و درصد داده‌های باقی‌مانده نیز نمایش داده می‌شود.

با در نظر گرفتن این توابع کمکی بالا، پیش پردازش داده‌ها شامل مراحل زیر است:

- انتخاب ویژگی‌ها: در این مرحله، ویژگی‌های مورد نظر برای آموزش و اعتبارسنجی تعیین می‌شوند. این ویژگی‌ها به‌عنوان مجموعه متغیرهای استفاده شده در مدل‌سازی داده‌ها محسوب می‌شوند. برای این منظور، چند نوع مجموعه ویژگی‌ها به نام‌های NORMAL_FEATURE، DELTA_VEGA و DELTA_VEGA_VANNA ایجاد شده‌اند.

حالات مختلف در نظر گرفته شده به شرح زیرند:

I. NORMAL_FEATURE: در این حالت، مجموعه ویژگی‌ها از دو ستون M0 و TAU0_IMPLVOL0 تشکیل می‌شود. به عبارت دیگر، این حالت دو ویژگی "M0" و "TAU0_IMPLVOL0" را استفاده می‌کند.

II. DELTA_VEGA: در این حالت، مجموعه ویژگی‌ها شامل سه ستون DELTA_BS، OVER_SQRT_TAU1_ و VEGA_N می‌شود. این حالت ویژگی‌های مربوط به "DELTA_BS" (دلتا بلک-شولز)، "OVER_SQRT_TAU_1" (برابر با معکوس مربعی از "TAU0") و "VEGA_N" (وگا) را استفاده می‌کند.

III. DELTA_VEGA_VANNA: در این حالت، مجموعه ویژگی‌ها از چهار ستون DELTA_BS، OVER_SQRT_TAU1_، VEGA_N و VANNA_N تشکیل شده است. به عبارت دیگر، این حالت ویژگی‌های مربوط به "DELTA_BS" (دلتا بلک-شولز)، "OVER_SQRT_TAU_1" (برابر با معکوس مربعی از "TAU0")، "VEGA_N" (وگا) و "VANNA_N" (وانا) را استفاده می‌کند.

بدین ترتیب در این بخش می‌توان مجموعه ویژگی‌های مد نظر خود را انتخاب کرد.

- استانداردسازی ویژگی‌ها: در این قسمت، مقادیر هر ویژگی استفاده شده در آموزش و اعتبارسنجی با استفاده از مدل استانداردسازی (STANDARDSCALER) به مقادیر استاندارد تبدیل می‌شوند. این عمل باعث می‌شود که ویژگی‌های ورودی با همان مقیاس قرار گیرند که یک نیاز مشترک در الگوریتم‌های یادگیری ماشین است.

- جداسازی داده‌ها: داده‌های آموزش و اعتبارسنجی با توجه به مقادیر ستون 'PERIOD0' به دو قسمت مختلف تقسیم می‌شوند. داده‌هایی که مقدار 'PERIOD0' آن‌ها برابر 0 است، به‌عنوان داده‌های آموزش استفاده می‌شوند و داده‌هایی که مقدار 'PERIOD0' آن‌ها برابر 1 است، به‌عنوان داده‌های اعتبارسنجی استفاده می‌شوند.

- پیش‌پردازش داده‌های تست: اطلاعات مربوط به داده‌های تست از یک فایل CSV خوانده شده و به صورت پیش‌پردازش شده‌ای که شامل تغییر نام ستون‌ها و ایجاد ویژگی‌های جدید است، آماده می‌شوند. سپس، نمونه‌هایی که مقدار ستون 'V1' آن‌ها موجود نیست، حذف می‌شوند.

این مراحل باعث می‌شود داده‌های آموزش، اعتبارسنجی و تست آماده‌ی استفاده در مدل شبکه عصبی شوند.

- گام 3

انتخاب بهترین هایپرپارمتر:

هدف از این بخش، تنظیم و آموزش یک شبکه عصبی است. در این بخش توابعی تعریف می‌شوند تا شبکه عصبی را با تنظیمات مختلف از هایپرپارامترها آموزش داده و آزمایش کنند، به ویژه با تمرکز بر پارامتر REG_ALPHA. بدین ترتیب کد روی مجموعه‌ای از مقادیر پیش‌فرض برای REG_ALPHA حرکت کرده و شبکه عصبی را با هر مقدار آموزش داده و آن را بر روی مجموعه اعتبارسنجی تست می‌کند. مقدار بهترین هایپرپارامتر REG_ALPHA بر اساس کمترین خطای میانگین مربعات اعتبارسنجی (MSE) نمایش داده می‌شود. در پایان، مقدار بهترین هایپرپارامتر و نمودار کاهش MSE طی هر EPOCH برای هر مقدار REG_ALPHA چاپ می‌شود.

تابع TRAIN_NEURAL_NETWORK هسته‌ای‌ترین تابع استفاده شده برای آموزش شبکه عصبی است. این تابع یک شبکه عصبی با استفاده از هایپرپارامترهای ارائه شده می‌سازد و آن را با داده‌های TRAIN آموزش می‌دهد.

تابع TEST_SELECTED_MODEL تابع اصلی استفاده شده برای آزمایش شبکه عصبی آموزش دیده روی داده‌های آزمایشی است. این تابع شبکه عصبی را با بهترین هایپرپارامترها پیدا شده می‌سازد، وزن‌های بهترین مدل را بارگیری می‌کند، مقادیر دلتا را برای مجموعه آزمایشی محاسبه می‌کند و مقادیر دلتای محاسبه شده را برمی‌گرداند.

- گام 4

آموزش مدل با استفاده از بهترین هایپرپارمتر:

این بخش از کد، مدل شبکه عصبی را با استفاده از بهترین هایپرپارامتری که از طریق تنظیم هایپرپارامتر به دست آمده‌است، روی داده‌های آموزش کامل آموزش می‌دهد.

ابتدا یک نمونه از کلاس HEDGENET ایجاد می‌شود که نماینده مدل شبکه عصبی برای مدلسازی HEDGING است. سپس ساخت BEST_MODEL با استفاده از هایپرپارامترها و تنظیمات مطلوب با استفاده از تابع BUILD_MODEL انجام می‌شود.

سپس متد FIT برای آموزش مدل با استفاده از ویژگی‌های انتخاب شده بر روی داده‌های آموزش فراخوانی می‌شود.

پس از آموزش، مدل BEST_MODEL حالا وزن‌ها و پارامترهای یادگرفته‌شده از داده‌های آموزش کامل با استفاده از بهترین هایپرپارامتر را دارد. این مدل آماده‌است که برای پیش‌بینی مقادیر دلتا بر روی داده‌های تست استفاده شود و عملکرد آن ارزیابی شود.

- گام 5

ارزیابی مدل روی داده‌های تست:

در این بخش از کد، مدل آموزش دیده بر روی داده‌های آزمایشی (DF_TEST) ارزیابی می‌شود. مراحل انجام شده به صورت زیر است:

ابتدا محاسبه دلتا با استفاده از بهترین مدل بوسیله‌ی متد CALCULATE_DELTA انجام می‌شود. برای محاسبه مقادیر دلتا برای داده‌های آزمایشی استفاده می‌شود. این متد داده‌های آزمایشی استاندارد شده و ویژگی‌های لازم برای محاسبه را به عنوان ورودی می‌گیرد.

سپس مقادیر دلتا محاسبه‌شده (DELTA_NN) برای محاسبه سود و زیان برای مدل شبکه عصبی (PNL_NN) و مدل بلک-شولز (PNL_BS) استفاده می‌شوند. سود و زیان بر اساس تفاضل مربعی بین قیمت گزینه با استفاده از دلتای پیش‌بینی‌شده و قیمت واقعی (حاصل از دلتای بلک شولز) محاسبه می‌شود.

گام 6:

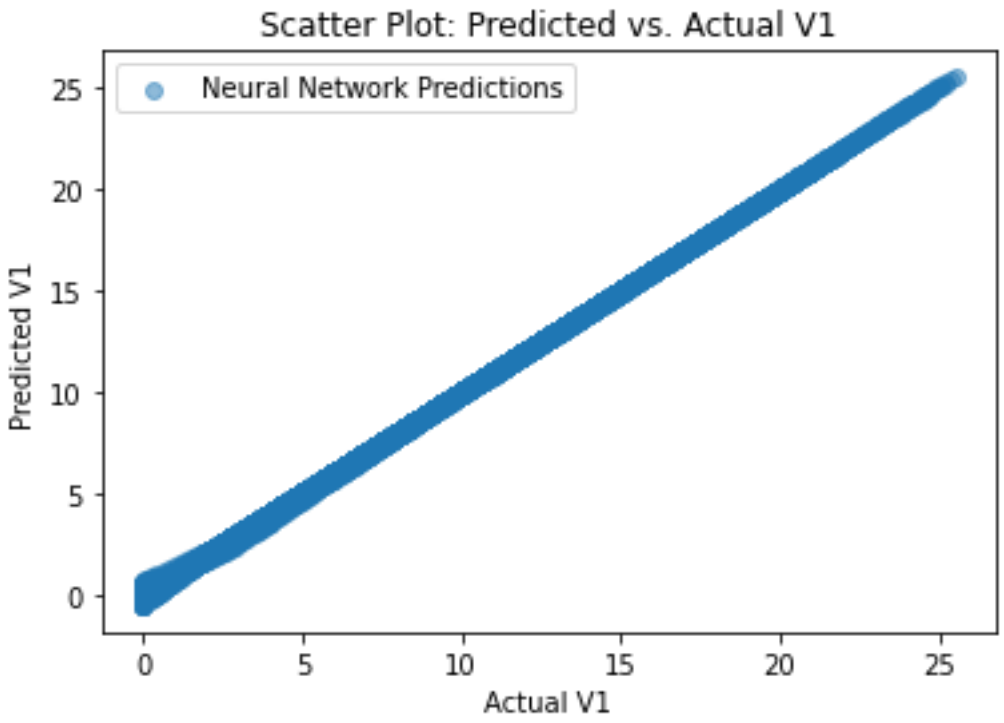
نتایج و مصورسازی:

در پایان HMSE محاسبه شده نهایی روی داده‌های تست گزارش شده و نمودارهای گوناگونی رسم شده اند که در ادامه هر آن‌ها را به همراه تفسیری از هر یک می‌آوریم.

1. نمودار پراکندگی (SCATTER PLOT)

این نمودار پیش‌بینی‌های شبکه عصبی برای ارزش گزینه‌ها را نسبت به ارزش‌های واقعی گزینه‌ها از داده‌های تست به تصویر می‌کشد.

در نمودار، هر نقطه نشان‌دهنده یک نقطه داده از داده‌های آزمایشی است که محور X آن نشان‌دهنده ارزش واقعی گزینه و محور Y نشان‌دهنده پیش‌بینی‌های مدل برای ارزش گزینه است..

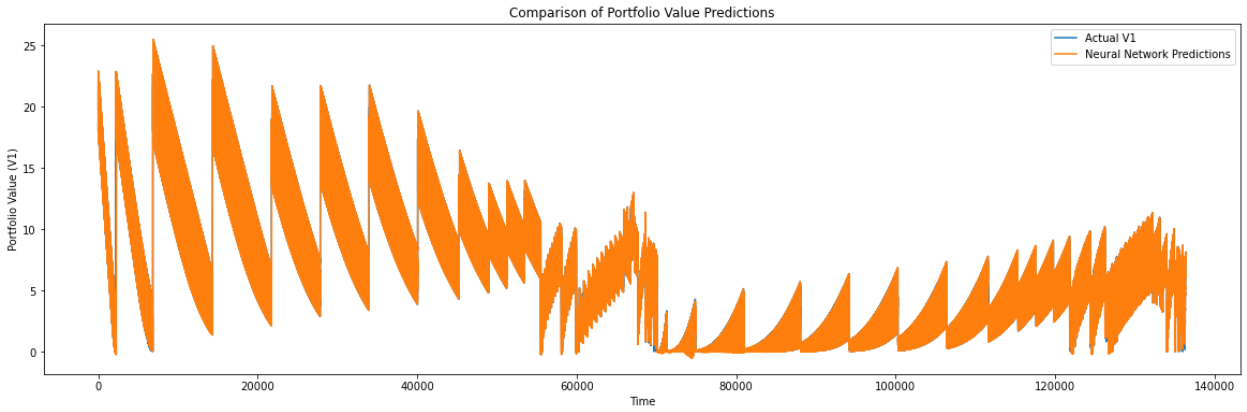


الگوهای نمودار پراکندگی به ما اطلاعاتی می‌دهد:

- با مشاهده نزدیکی نقاط به خط قطری (قطر نمودار)، می‌توانید دقت کلی مدل شبکه عصبی را ارزیابی کنید. نقاط نزدیک به خط قطری نشان‌دهنده پیش‌بینی‌های دقیق‌تر هستند، در حالی که نقاط دور از خط قطری نشان‌دهنده خطاهای پیش‌بینی می‌شوند.
- اگر نقاط به‌صورت خوشه‌ای در زیر یا بالای خط قطری تجمع داشته باشند، نشان‌دهنده این است که شبکه عصبی به‌طور سیستماتیک ارزش گزینه‌ها را به صورت کمتر یا بیش‌تری از واقعیت پیش‌بینی می‌کند. در نمودار خروجی این مدل چنین مشکلاتی دیده نمی‌شود.
- پراکندگی نقاط روی خط قطری می‌تواند محدوده‌ای از ارزش‌های گزینه‌ها که مدل به‌درستی پیش‌بینی می‌کند و محدوده‌ای که مدل دچار خطا می‌شود را نشان دهد.
- هر نقطه‌ای که به‌طور قابل توجهی از خط قطری فاصله بگیرد، به عنوان نقطه OUTLIER در نمودار مشخص می‌شود که نشان‌دهنده این است که مدل با یادگیری الگوی آن نقاط مشکل دارد.

2. نمودار زمان محور

در این بخش، نموداری رسم می‌شود که ارزش واقعی پرتفوی (V1) با پیش‌بینی‌های انجام‌شده توسط شبکه عصبی (NN) برای گزینه‌های مختلف طی زمان مقایسه می‌شوند.



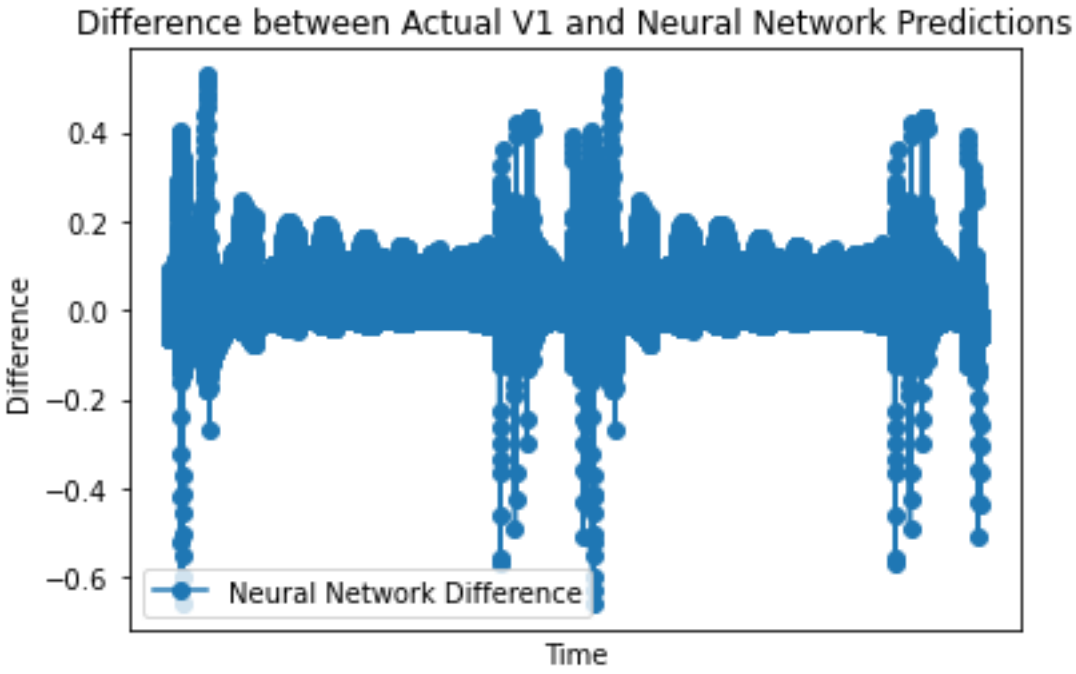
نکاتی که از این نمودار به‌دست می‌آید:

- با مقایسه ارزش واقعی پرتفوی (نمایش داده‌شده توسط خط 'ACTUAL V1') با پیش‌بینی‌های شبکه عصبی (نمایش داده‌شده توسط خط 'NEURAL NETWORK PREDICTIONS')، می‌توانیم ارزیابی کنیم که مدل چقدر توانسته با دقت پویایی‌های ارزش پرتفوی را به مدت زمان که پوشش می‌دهد، کنترل کند. با توجه تطابق مقدار پیشبینی با مقدار واقعی عملکرد مدل مناسب بوده است.
- از آنجایی که نمودار مربوط به 'NEURAL NETWORK PREDICTIONS' به‌خوبی نمودار 'ACTUAL V1' را دنبال می‌کند، نشان‌دهنده این است که شبکه عصبی پیش‌بینی‌های دقیقی انجام می‌دهد و مدل به‌طور موثری الگوهای زیربنایی ارزش پرتفوی را گرفته‌است.
- هر انحراف یا تاخیری بین نمودار 'NEURAL NETWORK PREDICTIONS' و 'ACTUAL V1' ممکن است نشان‌دهنده باشد که مدل با تاخیر به تغییرات ارزش پرتفوی واکنش نشان می‌دهد. همانطور که در نمودار مشاهده می‌شود در پیشبینی‌های صورت گرفته تاخیر واضحی وجود ندارد.
- تفاوت‌ها نوسانات میان دو خط می‌تواند نشان‌دهنده تفاوت‌ها در برآورد ولاتیلیتی باشد. اگر پیش‌بینی‌های شبکه عصبی نسبت به ارزش واقعی پرتفوی نوسان‌های معنادارتری نشان دهد، ممکن است نشان‌دهنده این باشد که مدل مقدار ولاتیلیتی را OVERESTIMATE می‌کند که نمودار حاصل به دلیل داشتن نوسانات مناسب و همگام با مقدار واقعی این مشکل را ندارد.

بدین ترتیب با استفاده از نمودار بالا، می‌توان به سادگی اطلاعاتی درباره پیش‌بینی‌های مدل شبکه عصبی برای ارزش پرتفوی به‌دست آمده در زمان‌های مختلف به‌دست آورده و عملکرد مدل را ارزیابی کرد.

3. نمودار تفاوت مقدار واقعی و پیشبینی‌شده

در این نمودار تفاوت بین مقادیر واقعی پرتفوی و پیش‌بینی‌های انجام‌شده توسط مدل شبکه عصبی (NN) رسم شده است. در این نمودار محور افقی نمایان‌گر زمان و محور عمودی نمایان‌گر تفاوت بین ارزش واقعی پرتفوی و پیش‌بینی‌های مدل است.



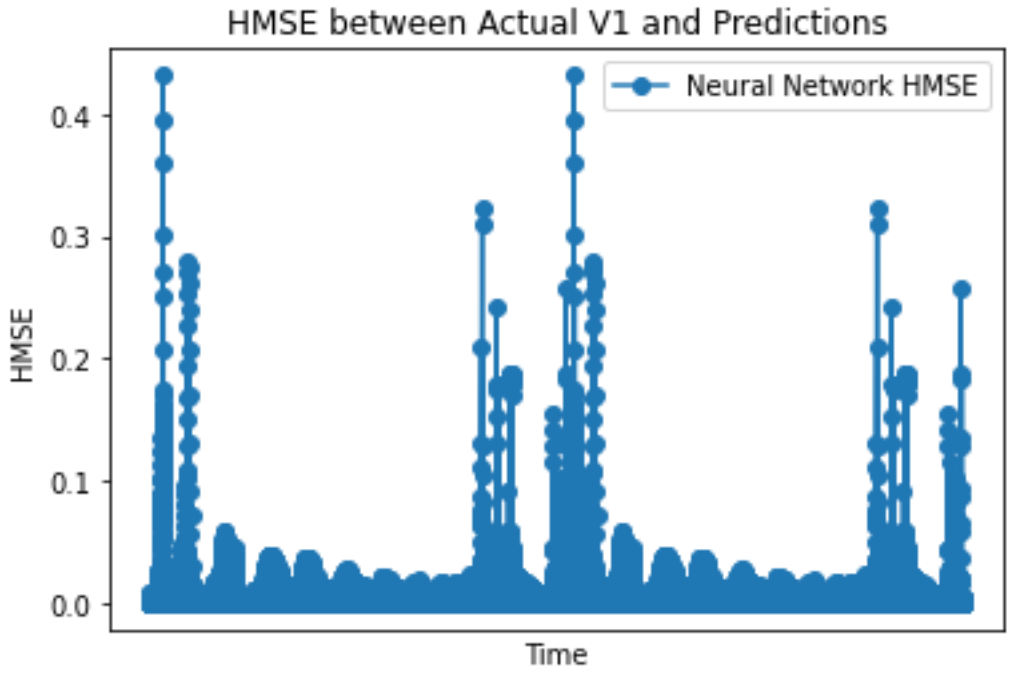
این نمودار اطلاعات مهمی در مورد پیش‌بینی‌های مدل شبکه عصبی برای ارزش پرتفوی به‌دست‌آمده در طول زمان ارائه می‌دهد. مهم‌ترین نکات به‌دست‌آمده از این نمودار عبارتند از:

- خطاهای محاسبه شده از پیشبینی به محور افقی نزدیکند این موضوع نشان می‌دهد که مدل با دقت خوبی توانسته الگوهای ارزش پرتفوی را برای زمان‌های مختلف کنترل کند.
- این نمودار به ما اجازه می‌دهد که عملکرد مدل را در طول زمان مشاهده کنیم و ببینیم که آیا در بازه‌های زمانی خاصی مدل عملکرد بهتری داشته است یا بهتر عمل نکرده است. در زمان‌هایی که خطا زیاد شده است مدل عملکرد ضعیف‌تری داشته است.
- هرگونه الگو یا روند در تفاوت بین ارزش واقعی پرتفوی و پیش‌بینی‌ها مشاهده می‌شود. الگوهای پایدار در تفاوت می‌توانند نشان‌دهنده شرایط بازار یا الگوهای داده‌ای باشند که مدل با آن‌ها مشکل دارد و نمی‌تواند پیش‌بینی دقیقی انجام دهد.

- هر نوسان بزرگ یا نقطه خارجی در نمودار می‌تواند نشان‌دهنده مواردی باشد که مدل قادر به درک برخی از رفتارهای بازاری یا مواجه با چالش‌ها در پیش‌بینی دقیق است.

4. نمودار HMSE

در این نمودار HMSE بین مقادیر واقعی پرتفوی و پیش‌بینی‌های انجام‌شده توسط مدل شبکه عصبی (NN) رسم شده است. محور افقی نمایانگر زمان و محور عمودی نمایانگر مقدار HMSE است.



نکات به‌دست‌آمده از این نمودار عبارتند از:

- این نمودار نشان می‌دهد که چقدر استراتژی HEDGING مدل شبکه عصبی در طول زمان خوب عمل می‌کند. مقدار کمتری از HMSE نشان‌دهنده این است که مدل با موفقیت ریسک را در پرتفوی کاهش می‌دهد.
- این نمودار به ما اجازه می‌دهد تا ببینیم چگونه عملکرد حفاظتی در بازه‌های زمانی مختلف تغییر می‌کند. اگر HMSE در طول زمان به شدت نوسان کند، این ممکن است نشان‌دهنده

این باشد که کارایی استراتژی HEDGINGبا شرایط بازار یا رفتار دارایی‌های زیربنایی تغییر می‌کند.

- نوسانات قابل توجه یا اوج‌های قابل مشاهده در HMSE ممکن است نشان‌دهنده دوره‌هایی باشد که استراتژی حفاظتی با مشکل روبرو شده و نتوانسته به طور موثر خطر را کاهش دهد. این دوره‌ها ممکن است با رویدادهای خاص مرتبط با دارایی‌های زیربنایی مرتبط باشد.

به‌طور خلاصه، این نمودارها اطلاعات ارزشمندی را درباره پیش‌بینی‌ها و عملکرد مدل شبکه عصبی در مقایسه با ارزش واقعی پرتفوی به‌دست می‌دهد.

5. در پایان مقدار کلی HMSE روی داده‌های تست نیز چاپ شده است.

```
# Print the overal HMSE
print(np.mean(hmse_nn))
```

```
0.0015342187035403355
```

مقدار تقریبی HMSE به دست آمده: 0.0015342187

از این مقدار برای مقایسه‌ی عملکرد شبکه‌ی عصبی با رگرسیون خطی استفاده می‌کنیم.

3. پیاده‌سازی و نتایج مدل‌های رگرسیون خطی

حال گام های پوشش ریسک با روش رگرسیون خطی را در کد به ترتیب توضیح می‌دهیم :

- گام 1

در این بخش تابع RUN_LINEAR_REGRESSION را تعریف می‌کنیم که انجام رگرسیون خطی را به صورت ROLLING (چرخشی) انجام می‌دهد.

این تابع ابتدا متغیرها و ساختارهای داده‌ای برای ذخیره ضرایب رگرسیون، انحرافات استاندارد معیار و مقادیر LEVERAGE را مقداردهی اولیه می‌کند. سپس دیتا را بر اساس گروه‌هایی به دسته‌های آموزش و تست تقسیم می‌کند.

سپس رگرسیون خطی بر داده‌های آموزش انجام می‌دهد. در اینجا، متغیر هدف بر اساس پارامتر TARGET_VARمحاسبه می‌شود. پارامتر TARGET کنترل می‌کند که متغیر هدف برای مدل رگرسیون خطی، تفاوت بین ارزش واقعی V1 و ارزش پیش‌بینی‌شده V1 توسط مدل بلک-شولز باشد یا تفاوت بین ارزش واقعی V1 و ارزش پیش‌بینی‌شده V1 بدون اعمال پوشش (NO-HEDGE) باشد.

سپس مدل رگرسیون اعمال شده و مقادیر هدف پیش‌بینی‌شده، باقیمانده‌ها و انحرافات استاندارد محاسبه می‌شوند.

اگر LEVERAGE برابر با TRUE باشد، یک رگرسیون اضافی روی زیرمجموعه‌ای از داده‌های آموزش انجام می‌شود و مقادیر LEVERAGE بر اساس ضرایب رگرسیون محاسبه می‌شوند.

سپس ضرایب رگرسیون، انحرافات استاندارد تناسب و مقادیر LEVERAGE (در صورت وجود) در DATAFRAME های مربوطه ذخیره می‌شوند. همچنین مقادیر DELTA برای داده‌های تست با استفاده از مدل رگرسیون پیش‌بینی می‌شود و در DATAFRAME مربوط به مقادیر DELTA ذخیره می‌شوند.

در نهایت، مقادیر DELTA در یک دیکشنری به نام RESULTS_DICT ذخیره می‌شود.

- گام 2

پیاده‌سازی توابع کمکی برای آماده‌سازی داده‌ها و بارگذاری و پیش پردازش داده‌ها مشابه گام 2 پیاده‌سازی شبکه‌های عصبی انجام می‌شود. به طور خلاصه در این بخش، ستون‌های نامربوط از داده‌ها را را حذف می‌کنیم ، FEATUREهای سفارشی را محاسبه و اضافه می‌کنیم و برچسب‌های داده را اختصاص می‌دهیم ، FEATUREها را استاندارد می‌کنیم و داده‌ها را آماده‌ی ورودی دادن به مدل می‌کنیم.

- گام 3

در این مرحله یک DICTIONARY خالی برای ذخیره نتایج نهایی به نام RESULTS_DICT ایجاد می‌کنیم. کلیدهای این دیکشنری نام استراتژی استفاده شده و مقادیر آن مقدار دلتای پیشبینی شده توسط مدل مربوطه است.

- گام 4

یک DATAFRAME جدید با ترکیب TRAIN و TEST ایجاد می‌کنیم و سپس یک آرایه با مقادیر صفر تولید و به عنوان دلتاهای استراتژی "NO_HEDGE" در دیکشنری نتایج ذخیره می‌کنیم.

- گام 5

در این گام مجموعه ویژگی‌های مختلفی را برای رگرسیون خطی در نظر گرفته‌ایم. با استفاده از این ترکیبات گوناگون ویژگی‌ها با صدا زدن تابع RUN_LINEAR_REGRESSION رگرسیون خطی را اجرا کرده و ضرایب حاصل را نمایش می‌دهیم. همزمان مقادیر دلتا برای هر روش را در دیکشنری نتایج ذخیره می‌کنیم تا بعدا از آن‌ها برای محاسبه ارزش نهایی سبد و نیز MSHE استفاده کنیم.

رگرسیون‌های خطی در نظر گرفته شده در این کد به شرح زیرند:

1. DELTA-ONLY ($\Delta = A * \Delta BS$)
2. DELTA-VEGA ($\Delta = A * \Delta BS + B * VBS$)
3. DELTA-GAMMA($\Delta = A * \Delta BS + B * \Gamma BS$)
4. DELTA-VANNA ($\Delta = A * \Delta BS + B * VABS$)
5. DELTA-GAMMA-VANNA ($\Delta = A * \Delta BS + B * \Gamma BS + C * VABS$)
6. DELTA-VEGA-GAMMA($\Delta = A * \Delta BS + B * VBS + C * \Gamma BS$)
7. DELTA-VEGA-GAMMA-VANNA ($\Delta = A * \Delta BS + B * VBS + C * \Gamma BS + D * VABS$)

8. DELTA-VEGA-VANNA(Δ = A * ΔBS + B * VBS +C * VABS)

9. VANNA-ONLY(Δ = A * VABS)

10. VEGA-ONLY(Δ = A * VBS)

11. GAMMA-ONLY(Δ = A * ΓBS)

12. HULL-WHITE LINEAR REGRESSION

13. RELAXED HULL-WHITE LINEAR REGRESSION

(توضیح مربوط به HULLWHITE در بخش تئوری آورده شده است.)

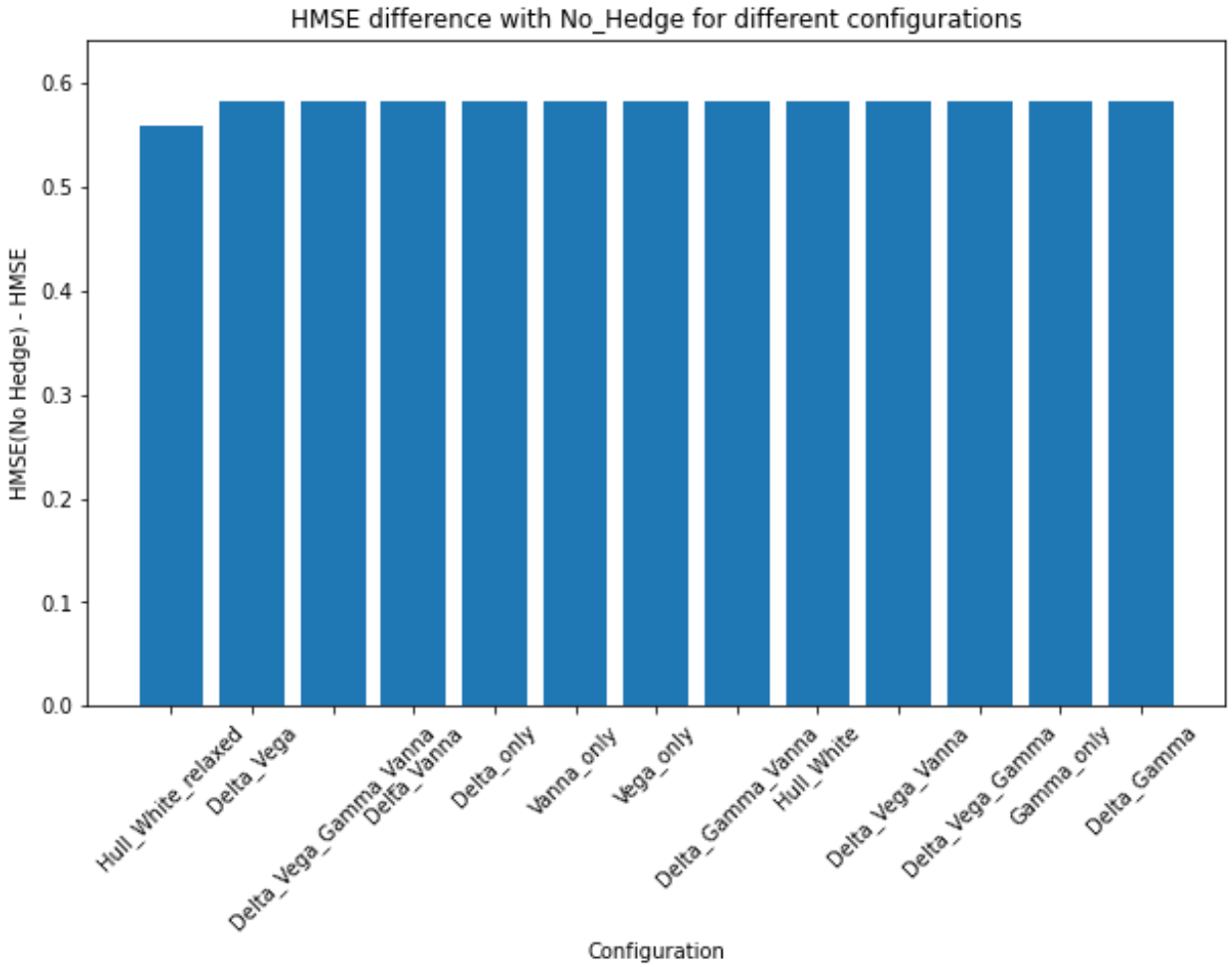
- گام 5

در پایان مقدار HMSE هر یک از مدل‌های رگرسیون خطی روی داده‌های تست محاسبه و چاپ شده است.

```
No_Hedge HMSE: 0.5841938456661125
BS HMSE: 0.0012855653441444655
Delta_only HMSE: 0.0013307338673194662
Delta_Vega HMSE: 0.0015386190552253362
Delta_Gamma HMSE: 0.0011941685820317206
Delta_Vanna HMSE: 0.001332793949772294
Delta_Gamma_Vanna HMSE: 0.0012500598246228765
Delta_Vega_Gamma HMSE: 0.0012065560389979333
Delta_Vega_Vanna HMSE: 0.0012410582133362386
Hull_White_relaxed HMSE: 0.024310208830454722
Hull_White HMSE: 0.0012479273029527339
Gamma_only HMSE: 0.0011941844000625062
Vega_only HMSE: 0.0012866928369154305
Vanna_only HMSE: 0.0012936925576464604
Delta_Vega_Gamma_Vanna HMSE: 0.0014297587646802143
```

همچنین مقادیر بهبود مدل رگرسیون خطی نسبت به روش NO-HEFGE (اختلاف HMSE این دو روش) چاپ شده و در یک نمودار ستونی به صورت SORT شده نمایش داده شده است.

```
Improvement over No_Hedge strategy
Hull_White_relaxed: 0.5598836368356578
Delta_Vega: 0.5826552266108872
Delta_Vega_Gamma_Vanna: 0.5827640869014323
Delta_Vanna: 0.5828610517163403
Delta_only: 0.5828631117987931
Vanna_only: 0.5829001531084661
Vega_only: 0.5829071528291971
Delta_Gamma_Vanna: 0.5829437858414896
Hull_White: 0.5829459183631598
Delta_Vega_Vanna: 0.5829527874527763
Delta_Vega_Gamma: 0.5829872896271147
Gamma_only: 0.5829996612660501
Delta_Gamma: 0.5829996770840808
```



بدین ترتیب می‌توانیم به آسانی عملکرد روش‌های مختلف را با هم مقایسه کنیم. هر مدلی که

همانطور که هدف مقاله نشان دادن این بود که شبکه عصبی پیاده‌سازی شده عملکرد بهتری از رگرسیون‌های خطی در کاهش ریسک ندارد، نتایج حاصل از شبیه‌سازی نیز این مساله را تایید

می‌کند. این نتیجه با مقایسه‌ی HMSE به دست آمده از مدل شبکه‌های عصبی و HMSE حاصل از رگرسیون‌های خطی محرز است. با توجه به مقادیر حاصل، مقدار HMSE شبکه عصبی از HMSE به دست آمده از اکثر مدل‌های مختلف شبکه‌های عصبی بیشتر است.