پروژه درس ریاضیات مالی

عنوان مقاله:

Solving high-dimensional optimal stopping time problems using deep learning

تابستان ۱۴۰۲

پرهام حق‌شناس ۴۰۱۲۱۰۵۸۳

مقدمه

امروزه بسیاری از مشتقات مالی، مانند اختیارات آمریکایی و برمودایی، به خریدار اجازه می‌دهند تا در صورت تمایل، زودتر از تاریخ سررسید از اختیار خود استفاده کند. قیمت‌گذاری چنین اختیاراتی نیازمند حل مسئله زمان توقف در بعدهای بالا است. این مسائل به دلیل ابعاد زیاد، به راحتی قابل حل‌کردن نیستند. در مقاله، الگوریتمی بر اساس یادگیری عمیق برای این دسته از مسائل ارائه شده است که علاوه بر به تخمین زمان توقف بهینه، تخمینی از قیمت را هم محاسبه می‌کند. در انتهای مقاله، این الگوریتم برای تعدادی مثال به کار گرفته شده است و نتایج به دست آمده از کارایی بالای این الگوریتم حکایت دارد.

ایده‌های اصلی الگوریتم

در این بخش، ایده‌های اصلی الگوریتم را در مسئله قیمت‌گذاری اختیارات آمریکایی بیان می‌کنیم. اگرچه الگوریتم پیشنهادی را می‌توان برای مسئله‌های دیگر با دینامیک متفاوت استفاده کرد. به خصوص برای اختیارات برمودایی که مسئله‌ای زمان گسسته است و نیازی به گسسته‌سازی فضای زمان‌های توقف ندارد.

قیمت‌گذاری اختیار آمریکایی

فرض کنید $T \in (0, \infty)$، $d \in \{1, 2, 3, \dots\}$، $(\Omega, \mathcal{F}, \mathbb{P})$ یک فضای احتمال باشد. قرار دهید: $X: [0, T] \times \Omega \to \mathbb{R}^d$

بردار وضعیت بازار (وضعیت d سهم) است و داریم:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t, \qquad X_0 = \xi, \qquad t \in [0, T]$$

که در آن W_t حرکت براونی استاندارد است.

فرض کنید $g: [0, T] \times \mathbb{R}^d \to \mathbb{R}$ تابع عایدی این اختیار باشد. اگر $\tau: \Omega \to [0, T]$ زمان توقف باشد، با توجه به استدلال‌های احتمال ریسک خنثی، قیمت اختیار از رابطه‌ی زیر به دست می‌آید:

$$sup\{E\big(g(\tau, X_\tau)\big)\}$$

گسسته‌سازی

برای گسسته‌سازی مسئله، نقاط بازه‌ی $[0, T]$ را به $N + 1$ نقطه‌ی

$$0 = t_0 < t_1 < \cdots < t_N = T$$

تقلیل می‌دهیم. داریم:

$$X_{t_{n+1}} = X_{t_n} + \int_{t_n}^{t_{n+1}} \mu(X_s)ds + \int_{t_n}^{t_{n+1}} \sigma(X_s)dW_s \quad , \qquad n \in \{0, 1, 2, \dots, N-1\}$$

و با احتمال تقریباً یک، تقریب زیر برقرار است:

$$X_{t_{n+1}} \approx X_{t_n} + \mu\big(X_{t_n}\big)(t_{n+1} - t_n) + \sigma\big(X_{t_n}\big)\big(W_{t_{n+1}} - W_{t_n}\big)$$

به همین دلیل فرآیند تصادفی $\chi = \big(\chi^{(1)}, \chi^{(2)}, \dots, \chi^{(d)}\big): \{0,1, \dots, N\} \times \Omega \to \mathbb{R}^d$ به شکل زیر تعریف می‌کنیم:

$$\chi_{n+1} = \chi_n + \mu(\chi_n)(t_{n+1} - t_n) + \sigma(\chi_n)\left(W_{t_{n+1}} - W_{t_n}\right)$$

و داریم:

$$sup\{E\left(g(\tau, \mathrm{X}_\tau)\right): \tau: \Omega \to [0, T]\ stopping\ time\} \approx$$
$$sup\{E(g(\mathrm{t}_\tau, \chi_\tau\ )): \tau: \Omega \to \{0, 1, \dots, N\}\ stopping\ time\}$$

با اینکه عایدی در حالت با دانستن χ مشخص می‌شود، مقدار آن را به بردار χ اضافه می‌کنیم زیرا فرآیند یادگیری بهبود می‌یابد:

$$Z_n = \left(\chi_n, g(t_n, \chi_n)\right)$$

و از این پس با فرآیند تصادفی Z_n کار می‌کنیم.

فرمول‌بندی زمان توقف

برای هر $n\ =\ 0,1, \dots, N$، قرار دهید $U_n: (\mathbb{R}^{d+1})^{n+1} \to \{0,1\}$ که U_n تابع مشخصه‌ی $\tau = n$ است و در نتیجه برای هر $\omega \in \Omega$ داریم:

$$\sum_{i=0}^{n} U_n\left(Z_0(\omega), \dots, Z_n(\omega)\right) =\ 1$$

به علاوه:

$$\sum_{i=0}^{n} nU_n\left(Z_0(\omega), \dots, Z_n(\omega)\right) =\ \tau$$

همینطور فرض کنید $U_{n,\tau}$ها توابع متناظر با حالتی باشد که در زمان τ توقف کنیم.

در قسمت بعد، برای تخمین زدن U_nهای بهینه، از شبکه‌های عصبی و یادگیری عمیق استفاده می‌کنیم.

ساختار شبکه عصبی

توابع $u_n: \mathbb{R}^{d+1} \to \{0,1\}$ را طوری در نظر بگیرید که در روابط زیر صدق کنند (اگر هر مقدار صفر و یک قابل قبول بود، مقدار تابع را صفر در نظر می‌گیریم.):

$$U_n(z_1, z_2, \dots, z_n) = max\{u_n(z_n), n + 1 - N\}\left[1 - \sum_{k=0}^{n-1} U_k(z_1, z_2, \dots, z_k)\right], n\ =\ 0, 1, 2, \dots, N$$

فرض کنید $\theta \in \mathbb{R}^\nu$ بردار وزن‌های شبکه‌های عصبی باشد، برای $n\ =\ 0, 1, \dots, N$، فرض کنید $u_{n,\theta}: \mathbb{R}^{d+1} \to (0,1)$

تخمین به دست آمده برای تابع u_n با استفاده از شبکه‌ی عصبی با معماری زیر باشد (برای تخمین زدن هر یک از توابع از یک شبکه‌ی عصبی مجزا استفاده می‌کنیم که ورودی آن فرآیند تصادفی Z_n است.):

- لایه ورودی شامل $d + 1$ نورون و بایاس (تابع ثابت ۱)
- دو لایه مخفی هر کدام شامل l نورون و بایاس
- لایه خروجی با یک نورون

در این صورت هر یک از N شبکه‌ی عصبی شامل $l(d+2)$ یال بین دو لایه‌ی اول، $l(l+1)$ بین لایه‌های مخفی و $l + 1$ یال بین دو لایه‌ی آخر است و در نتیجه داریم: $\nu = N(l^2 + (d+4)l + 1)$

در مثال‌های بررسی شده، تابع فعالسازی نورون‌های لایه‌ی خروجی را $\mathcal{L}(x) = \frac{e^x}{e^x+1}$ و نورون‌های لایه‌های مخفی را $max\{x, 0\}$ در نظر می‌گیریم.

اجزای دیگر این شبکه‌ها یعنی تابع هدف و الگوریتم تغییر وزن‌ها را در بخش‌های بعدی بیان می‌کنیم.

فرمول‌بندی تابع هدف

همانطور که در بخش‌های قبل ذکر شد، قیمت ورقه‌ی مشتقه عبارت است از:

$$sup\{E(g(\mathrm{t}_\tau, \chi_\tau)): \tau: \Omega \to \{0, 1, \dots, N\}\ stopping\ time\}$$

داریم:

$$g(\mathrm{t}_\tau, \chi_\tau) = \sum_{n=0}^{N} U_{n,\tau}(Z_1, Z_2, \dots, Z_n) g(t_n, \chi_n) \approx \sum_{n=0}^{N} U_{n,\theta}(Z_1, Z_2, \dots, Z_n) g(t_n, \chi_n)$$

پس تابع زیر را به عنوان تخمینی از تابع هدف در نظر می‌گیریم:

$$sup\{E(\sum_{n=0}^{N} U_{n,\theta}(Z_1, Z_2, \dots, Z_n) g(t_n, \chi_n)): \tau: \Omega \to \{0, 1, \dots, N\}\ stopping\ time\}$$

البته چون محاسبه این تابع هدف هم از نظر محاسباتی چالش‌برانگیز است، برای محاسبه آن هم از تخمین‌گرهای تصادفی ناااریب (روش‌های مونت کارلو) استفاده می‌کنیم.

روش کاهش گرادیان تصادفی برای بهینه‌سازی وزن‌های شبکه‌ی عصبی

همانطور که در بخش قبلی ذکر شد، برای رسیدن به نقطه‌ی بهینه برای تابع هدف

$$sup\{E(\sum_{n=0}^{N} U_{n,\theta}(Z_1, Z_2, \dots, Z_n) g(t_n, \chi_n)): \tau: \Omega \to \{0, 1, \dots, N\}\ stopping\ time\}$$

الگوریتم کاهش گرادیان تصادفی را برای تابع تصادفی

$$\sum_{n=0}^{N} U_{n,\theta}\big(Z_1(\omega), Z_2(\omega), \dots, Z_n(\omega)\big) g(t_n, \chi_n)$$

استفاده می‌کنیم و با استفاده از آن وزن‌های شبکه‌های عصبی را به روزرسانی می‌کنیم. در این صورت دنباله‌ای از وزن‌ها مانند دنباله‌ی زیر به دست می‌آید و در نهایت فرآیند یادگیری به پایان می‌رسد و وزن‌های نهایی را می‌توان برای به دست آوردن قیمت اختیارات استفاده کرد:

$$\{\Theta_m = \left(\Theta_m^{(1)}, \Theta_m^{(2)}, \dots, \Theta_m^{(\nu)}\right)\}_{m=1}^{M}$$

فرمول‌بندی تخمین‌گرهای زمان توقف بهینه و قیمت برای اختیارات آمریکایی

با استفاده از الگوریتم قسمت قبل، برای N, ν, M به اندازه‌ی کافی بزرگ، مقدار

$$E\left(\sum_{n=0}^{N} U_{n,\Theta_M}(Z_1, Z_2, \dots, Z_n) g(t_n, \chi_n)\right)$$

که خود با استفاده از روش مونت کارلو تخمین زده می‌شود، تخمین‌گری برای قیمت اختیار آمریکایی و مقدار

$$\sum_{i=0}^{n} n U_{n,\Theta_M}\big(Z_0(\omega), \dots, Z_n(\omega)\big)$$

تخمین‌گری برای زمان توقف بهینه است. اما مقدار بالا لزوماً در بازه‌ی $[0, N]$ قرار نمی‌گیرد. به علاوه، لزومی ندارد که عایدی که با این زمان توقف به دست می‌آید، برابر با تخمین‌گر قیمت (عایدی) باشد. به همین دلیل، زمان توقف بهینه برای بردار وزن θ را با تخمین‌گر زیر به دست می‌آوریم:

$$\tau_\theta = min\,\{n \in \{0,1,\dots,n\}: \sum_{k=0}^{n} U_{k,\theta}(Z_0, Z_1, \dots, Z_k) \geq 1 - U_{n,\theta}(Z_0, Z_1, \dots, Z_n)\}$$

به زبان ساده‌تر، زمان توقف بهینه، اولین n است که مجموع توابع مشخصه $\tau = 0$ تا $\tau = n$ از یک بزرگتر یا با آن مساوی شود. همینطور امید عایدی به دست آمده با استفاده از این زمان توقف را به عنوان تخمین‌گر قیمت در نظر می‌گیریم. البته باز هم تکرار می‌کنیم که امید ریاضی با استفاده از روش مونت کارلو تخمین زده می‌شود.

جزییات الگوریتم

فرمول‌بندی الگوریتم در یک حالت خاص

در این بخش الگوریتم را برای حل مسئله‌ی قیمت‌گذاری اختیار آمریکایی که در ابتدا مطرح شد، بیان می‌کنیم. همچنین برای تغییر وزن‌ها از الگوریتم کاهش گرادیان ساده با نرخ یادگیری ثابت $\gamma \in (0, \infty)$ استفاده می‌کنیم.

فرض کنید $T, \gamma \in (0, \infty)$، d, N, l اعداد طبیعی، $\nu = N(l^2 + l(d+4) + 1)$، $\mu: \mathbb{R}^d \to \mathbb{R}^d$، $\sigma: \mathbb{R}^d \to \mathbb{R}^{d\times d}$ و $g: [0,T] \times \mathbb{R}^d \to \mathbb{R}$ توابع اندازه‌پذیر بورل، $(\Omega, \mathcal{F}, \mathbb{P})$ یک فضای احتمال، برای عدد طبیعی m، $\xi^m: \Omega \to \mathbb{R}^d$ متغیرهای تصادفی مستقل، $W^m: [0,T] \times \Omega \to \mathbb{R}^d$ حرکت‌های براونی استاندارد مستقل و مستقل از ξ^mها باشند. همینطور فرض کنید $t_0, t_1, \dots, t_N \in [0,T]$ اعداد حقیقی باشند که $0 = t_0 < t_1 < \dots < t_N = T$، برای عدد طبیعی m

$\chi^m = \left(\chi^{m,(1)}, \dots, \chi^{m,(d)}\right): \{0,1,\dots,N\} \times \Omega \to \mathbb{R}^d$ و $Z^m: \{0,1,\dots,N\} \times \Omega \to \mathbb{R}^{d+1}$ فرآیندهای تصادفی باشند که برای عدد طبیعی m و n که $n \in \{0, 1, \dots, N-1\}$ داشته باشیم $\chi_0^m = \xi^m$،

$$\chi_{n+1}^m = \chi_n^m + \mu(\chi_n^m)(t_{n+1} - t_n) + \sigma(\chi_n^m)\left(W_{t_{n+1}}^m - W_{t_n}^m\right)$$

و

$$Z_n^m = \left(\chi_n^{m,(1)}, \dots, \chi_n^{m,(d)}, g(t_n, \chi_n^m)\right)$$

فرض کنید برای هر عدد طبیعی k، $\mathcal{L}_k: \mathbb{R}^k \to \mathbb{R}^k$ تابعی باشد که برای هر $x = (x_1, x_2, \dots, x_k) \in \mathbb{R}^k$ داشته باشیم:

$$\mathcal{L}_k(x) = \left(\frac{exp(x_1)}{exp(x_1) + 1}, \dots, \frac{exp(x_k)}{exp(x_k) + 1}\right)$$

و برای هر $\theta = (\theta_1, \dots, \theta_\nu) \in \mathbb{R}^\nu$، $v \in \mathbb{N}_0$، $k, j \in \mathbb{N}$ که $v + k(j+1) \le \nu$ فرض کنید $A_{k,j}^{\theta,v}: \mathbb{R}^j \to \mathbb{R}^k$ تابعی باشد که در رابطه‌ی زیر صدق می‌کند:

$$A_{k,j}^{\theta,v}(x) = \left(\theta_{v+kj+1} + \left[\sum_{i=1}^{j} x_i \theta_{v+i}\right], \dots, \theta_{v+kj+k} + \left[\sum_{i=1}^{j} x_i \theta_{v+(k-1)j+i}\right]\right)$$

برای هر $n \in \{0, 1, \dots, N-1\}$، $u_{n,\theta}: \mathbb{R}^{d+1} \to (0,1)$ (توابع خروجی شبکه‌های عصبی) را چنین تعریف می‌کنیم:

$$u_{n,\theta} = \mathcal{L}_1 \circ A_{1,l}^{\theta, n\left(l^2 + l(d+4) + 1\right) + l(l+d+3)} \circ \mathcal{L}_l \circ A_{l,l}^{\theta, n\left(l^2 + l(d+4) + 1\right) + l(d+2)} \circ \mathcal{L}_l \circ A_{l,d+1}^{\theta, n\left(l^2 + l(d+4) + 1\right)}$$

و $U_{n,\theta}: (\mathbb{R}^{d+1})^{n+1} \to (0,1)$ را چنین تعریف می‌کنیم:

$$U_{n,\theta}(z_0, z_1, \dots, z_n) = max\{u_{n,\theta}(z_n), n + 1 - N\}\left[1 - \sum_{k=0}^{n-1} U_{k,\theta}(z_0, z_1, \dots, z_k)\right]$$

تابع هدف $\phi^m: \mathbb{R}^\nu \times \Omega \to \mathbb{R}$ برای هر m طبیعی عبارت است از:

$$\phi^m(\theta, \omega) = \sum_{n=0}^{N} \left[U_{n,\theta}\left(Z_0^m(\omega), \dots, Z_n^m(\omega)\right) g\left(t_n, \chi_n^m(\omega)\right)\right]$$

و گرادیان آن $\Phi^m: \mathbb{R}^\nu \times \Omega \to \mathbb{R}^\nu$ عبارت است از:

$$\Phi^m(\theta, \omega) = (\nabla_\theta \phi^m)(\theta, \omega)$$

فرض کنید $\Theta: \mathbb{N}_0 \times \Omega \to \mathbb{R}^\nu$ فرآیند تصادفی باشد که برای هر $m \in \mathbb{N}$ داشته باشیم:

$$\Theta_m = \Theta_{m-1} + \gamma \Phi^m(\Theta_{m-1})$$

و زمان توقف $\tau_{j,\theta}: \Omega \to \{0, 1, \dots, N\}$ را برای هر $j \in \mathbb{N}$ را چنین تعریف می‌کنیم:

$$\tau_{j,\theta} = min\{n: \sum_{k=0}^{n} U_{k,\theta}\left(Z_0^j, \dots, Z_k^j\right) \ge 1 - U_{n,\theta}\left(Z_0^j, \dots, Z_n^j\right)\}$$

فرض کنید زمان یادگیری (تعداد تغییرات وزن‌های شبکه‌ی عصبی) $M \in \mathbb{N}$ باشد. در اینصورت برای $J \in \mathbb{N}$ متغیر تصادفی زیر تخمین‌گر مونت کارلو برای قیمت خواهد بود:

$$\frac{1}{J}\sum_{j=1}^{J} g\left(t_{\tau_{m+j},\Theta_M}, \chi^{M+j}_{\tau_{M+j},\Theta_M}\right)$$

و τ_{M+j}, Θ_M تخمین‌گر زمان توقف بهینه خواهد بود.

فرمول‌بندی الگوریتم در حالت کلی

فرض کنید $T \in (0,\infty)$، $d, N, M, \varrho, \varsigma, \nu$ اعداد طبیعی، $g : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ توابع اندازه‌پذیر بورل، $(\Omega, \mathcal{F}, \mathbb{P})$ یک فضای احتمال، $t_0, t_1, \dots, t_N \in [0,T]$ اعداد حقیقی باشند که $0 = t_0 < t_1 < \cdots < t_N = T$، برای $m \in \mathbb{N}_0$ و عدد طبیعی j، $\chi^{m,\mathrm{j}} = \left(\chi^{m,\mathrm{j},(1)}, \dots, \chi^{m,\mathrm{j},(d)}\right) : \{0,1,\dots,N\} \times \Omega \to \mathbb{R}^d$ و $Z^{m,\mathrm{j}} : \{0,1,\dots,N\} \times \Omega \to \mathbb{R}^{d+1}$ فرآیندهای تصادفی باشند که

$$Z^{m,\mathrm{j}}_n = \left(\chi^{m,\mathrm{j},(1)}_n, \dots, \chi^{m,\mathrm{j},(d)}_n, g\left(t_n, \chi^{m,\mathrm{j}}_n\right)\right)$$

و برای هر $\theta = (\theta_1, \dots, \theta_\nu) \in \mathbb{R}^\nu$ و $s \in \mathbb{R}^\varsigma$ و $n \in \{0, 1, \dots, N\}$، $\mathrm{u}^{\theta,s}_n : \mathbb{R}^{d+1} \to (0,1)$ (توابع خروجی شبکه‌های عصبی) باشند. $U^{\theta,s}_n\{: (\mathbb{R}^{d+1})^{n+1} \to (0,1)$ را چنین تعریف می‌کنیم:

$$U^{\theta,s}_n(z_0, z_1, \dots, z_n) = max\{\mathrm{u}^{\theta,s}_n(z_n), n+1-N\}\left[1 - \sum_{k=0}^{n-1} U^{\theta,s}_{\mathrm{k}}(z_0, z_1, \dots, z_k)\right]$$

تابع هدف $\phi^{m,s} : \mathbb{R}^\nu \times \Omega \to \mathbb{R}$ برای هر m طبیعی و $s \in \mathbb{R}^\varsigma$ عبارت است از:

$$\phi^{m,s}(\theta,\omega) = \frac{1}{J_m}\sum_{j=1}^{J_m}\sum_{n=0}^{N}\left[U^{\theta,s}_n\left(Z^{m,\mathrm{j}}_0(\omega), \dots, Z^{m,\mathrm{j}}_n(\omega)\right) g\left(t_n, \chi^{m,\mathrm{j}}_n(\omega)\right)\right]$$

و گرادیان آن $\Phi^{m,s} : \mathbb{R}^\nu \times \Omega \to \mathbb{R}^\nu$ عبارت است از:

$$\Phi^{m,s}(\theta,\omega) = (\nabla_\theta \phi^{m,s})(\theta,\omega)$$

فرض کنید $\mathcal{S} : \mathbb{R}^\varsigma \times \mathbb{R}^\nu \times (\mathbb{R}^{d+1})^{\{0,1,\dots,N-1\}\times N} \to \mathbb{R}^\varsigma$ یک تابع، برای هر $m \in \mathbb{N}$، $\Psi_m : \mathbb{R}^\varrho \times \mathbb{R}^\nu \to \mathbb{R}^\varrho$ و $\psi_m : \mathbb{R}^\varrho \to \mathbb{R}^\nu$ تابع، $\mathbb{S} : \mathbb{N}_0 \times \Omega \to \mathbb{R}^\varsigma$، $\Xi : \mathbb{N}_0 \times \Omega \to \mathbb{R}^\varrho$ و $\Theta : \mathbb{N}_0 \times \Omega \to \mathbb{R}^\nu$ فرآیندهای تصادفی باشد که برای هر $m \in \mathbb{N}$ داشته باشیم:

$$\mathbb{S}\_m = \mathcal{S}(\mathbb{S}_{m-1}, \Theta_{m-1}, (Z^{m,j}_n)_{\{(n,j)\in\{0,1,\dots,N-1\}\times\mathbb{N})})$$

$$\Xi_m = \Psi_m\left(\Xi_{m-1}, \Phi^{m,\mathbb{S}_m}(\Theta_{m-1})\right)$$

$$\Theta_m = \Theta_{m-1} + \psi_m(\Xi_m)$$

و زمان توقف $\tau^{j,\theta,s} : \Omega \to \{0,1,\dots,N\}$ را برای هر $j \in \mathbb{N}$ را چنین تعریف می‌کنیم:

$$\tau^{j,\theta,s} = min\{n: \sum_{k=0}^{n} U^{\theta,s}_{\mathrm{k}}\left(Z^{0,j}_0, \dots, Z^{0,j}_k\right) \geq 1 - U^{\theta,s}_n\left(Z^{0,j}_0, \dots, Z^{0,j}_n\right)\}$$

فرض کنید زمان یادگیری (تعداد تغییرات وزن‌های شبکه‌ی عصبی) $M \in \mathbb{N}$ باشد. در اینصورت برای $J_0 \in \mathbb{N}$ متغیر تصادفی زیر تخمین‌گر مونت کارلو برای قیمت خواهد بود:

$$\frac{1}{J_0}\sum_{j=1}^{J_0} g\left(t_{\tau^{j,\Theta_M,\mathbb{S}_M}}, \chi^{0,j}_{\tau^{j,\Theta_M,\mathbb{S}_M}}\right)$$

و $\tau^{j,\Theta_M,\mathbb{S}_M}$ تخمین‌گر زمان توقف بهینه خواهد بود.

مثال‌های عددی

فرمول‌بندی چهارچوب کلی

در فرمول‌بندی الگوریتم در حالت کلی قرار دهید، $\zeta_1 = 0.9$، $\zeta_2 = 0.999$، $\epsilon \in (0,\infty)$، $(\gamma_m)_{m\in\mathbb{N}} \subseteq (0,\infty)$، $\xi =$ $(\xi_1,\dots,\xi_d) \in \mathbb{R}^d$، $\mathfrak{F} = (\mathfrak{F}_t)_{[0,T]}$ فیلتریشنی روی فضای احتمال $(\Omega,\mathcal{F},\mathbb{P})$ باشد که شرایط معمول را دارد، برای $m \in$ $\mathbb{N}_0$ و $j \in \mathbb{N}$، $W^{m,j} = \left(W^{m,j,(1)},\dots,W^{m,j,(d)}\right)$ حرکت‌های براونی استاندارد مستقل، $\mu: \mathbb{R}^d \to \mathbb{R}^d$ و $\sigma: \mathbb{R}^d \to$ $\mathbb{R}^{d\times d}$ توابع پیوسته و لیپشیتز، $X = \left(X^{(1)},\dots,X^{(d)}\right): [0,T]\times\Omega \to \mathbb{R}^d$ فرآیند تصادفی باشد که تقریباً مطمئناً داشته باشیم:

$$X_t = \xi + \int_0^t \mu(X_s)ds + \int_0^t \sigma(X_s)dW_s^{0,1}, t \in [0,1]$$

همینطور $\varrho = 2\nu$، $\Xi_0 = 0$ و $t_n = \frac{nT}{N}$. برای هر $m \in \mathbb{N}$، $x,y,\eta \in \mathbb{R}^\nu$ قرار دهید:

$$\Psi_m(x,y,\eta) = \left(\zeta_1 x + (1-\zeta_1)\eta, \zeta_2 y + (1-\zeta_2)(\eta_1^2,\dots,\eta_\nu^2)\right)$$

و

$$\psi_{\mathrm{m}}(x,y) = \left(\left[\sqrt{\frac{|\mathrm{y}_1|}{1 - \zeta_2^{\mathrm{m}}}} + \varepsilon\right]^{-1} \frac{\gamma_{\mathrm{m}}\mathrm{x}_1}{1 - \zeta_1^{\mathrm{m}}}, \dots, \left[\sqrt{\frac{|\mathrm{y}_\nu|}{1 - \zeta_2^{\mathrm{m}}}} + \varepsilon\right]^{-1} \frac{\gamma_{\mathrm{m}}\mathrm{x}_\nu}{1 - \zeta_1^{\mathrm{m}}}\right)$$

ساختار بهینه‌ساز ADAM باشد که در شبکه‌های عصبی متشکل از دو لایه‌ی مخفی با $d + 50$ نورون استفاده شده است.

ملاحظه‌ای درباره‌ی مشکلات سخت‌افزاری شبیه‌سازی‌ها

مثال‌های عددی با پکیج tensorflow در محیط پایتون اجرا شده‌اند. محاسبات مربوط به شبکه‌های عصبی معمولاً با GPU اجرا می‌شوند و اجرای آن‌ها با CPU به زمان زیادی نیاز احتیاج دارد و در اکثر مثال‌های مقاله، اجرای هر کد با CPU بیش از یک روز طول می‌کشید. از آنجا که بنده به لپ‌تاپ یا سیستمی با GPU دسترسی نداشتم، کدها را در سایت Colabratory گوگل اجرا کرده‌ام که امکان استفاده از GPU را فراهم می‌کند. اما چون در ایران تنها نسخه‌ی رایگان این سایت در دسترس است، محدودیت‌هایی برای فضای

GPU و زمان اجرا وجود دارد. برای کاهش زمان اجرا، تعداد شبیه‌سازی‌ها در بعضی از مثال‌ها کمتر از تعداد شبیه‌سازی‌هایی است که در مقاله انجام شده است اما در تمام موارد نتایج به دست آمده مشابه نتایج مقاله است که نشان از صحت نتایج کدهای اجرا شده در این پروژه دارد. در مثال آخر، فضای GPU مورد نیاز برای اجرای کدها به دلیل ابعاد بالای مسئله، از محدودیت‌های سایت Colab فراتر بود و امکان اجرای کد برای ابعاد بالا وجود نداشت. لازم به ذکر است که GPU استفاده شده در مقاله، NVIDIA 2080ti است که تقریبا دو برابر سریع‌تر از GPU سایت Colab است و با توجه به این ملاحظه، زمان اجرای شبیه‌سازی‌ها در سایت Colab معقول است که نشان می‌دهد تفاوت در زمان اجرا واقعاً مربوط به سخت‌افزار است و تفاوت زیادی بین زمان‌های مقاله و شبیه‌سازی‌های انجام شده در این پروژه وجود ندارد. جزئیات محدودیت‌های سخت‌افزاری در هر مثال به طور جداگانه ذکر شده است. نتایج شبیه‌سازی‌ها در فایل‌های اکسل و کدها در فایل‌های پایتون آمده است.

مثال ٤.٣.١.١

در فرمول‌بندی چهار چوب کلی، فرض کنید

$J_{\mathrm{m}} = J_0 = 4096000$ ،$M = 500$ ،$N = 2$ ،$T = 1$ ،$\chi = 95$ ،$K = 90$ ،$\beta = 0.3$ ، $r = 0.03$
،$\epsilon = 10^{-8}$ ،8192

$$\gamma_m = 5\left[10^{-2}\mathbf{1}_{[1,100]}(m) + 10^{-3}1_{(100,300]}(m) + 10^{-4}\mathbf{1}_{(300,\infty)}(m)\right]$$

$$Q = GG^t \text{ ،} Q = 0.1\,\mathbf{1}_d\mathbf{1}_d^{\mathrm{t}} + 0.9I_d$$

$$\chi_n^{m-1,j} = GW_{t_n}^{m-1,j}$$

و

$$g(s,x) = e^{-rs}max\{K - exp\left([r - 0.5\beta^2]s + \frac{\beta\sqrt{10}}{\sqrt{d(d+9)}}[\mathrm{x}_1 + \cdots + \mathrm{x}_{\mathrm{d}}]\right)\chi, 0\}$$

در این مثال به جای ١٠ بار، ١ بار قیمت را به دست آورده‌ایم.

مثال ٤.٣.١.٢

در فرمول‌بندی چهار چوب کلی، فرض کنید

،$N = 50$ ،$T = 1$ ،$\chi = 40$ ،$K = 40$ ،$\beta = 0.4$ ، $r = 0.06$

$$M = 1500\,\mathbf{1}_{[1,50]}(d) + 1800\,\mathbf{1}_{(50,100]}(d) + 3000\,\mathbf{1}_{(100,\infty)}(d)$$

،$\epsilon = 10^{-3}$ ،$J_{\mathrm{m}} = 8192\,\mathbf{1}_{[1,50]}(d) + 4096\,\mathbf{1}_{(50,100]}(d) + 2048\,\mathbf{1}_{(100,\infty)}(d)$ ،$J_0 = 4096000$ ،

$$\gamma_m = 5\left[10^{-2}\mathbf{1}_{[1,\mathrm{M}/_3]}(m) + 10^{-3}1_{(\mathrm{M}/_3,2\mathrm{M}/_3]}(m) + 10^{-4}\mathbf{1}_{(2\mathrm{M}/_3,\infty)}(m)\right] \text{،}$$

$$\chi_n^{m-1,j} = W_{t_n}^{m-1,j}$$

و

$$g(s,x) = e^{-rs} max\{K - exp\left([r-0.5\beta^2]s + \frac{\beta}{\sqrt{d}}[\mathrm{x}_1 + \cdots + \mathrm{x}_\mathrm{d}]\right)\chi, 0\}$$

در این مثال به جای ۱۰ بار، ۱ بار قیمت را به دست آورده‌ایم.

مثال ٤.٤.١.١

در فرمول‌بندی چهار چوب کلی، فرض کنید $r = 0.05$ ، $\delta = 0.1$، $\beta = 0.2$، $K = 100$، $T = 3$، $N = 9$، $M = 3000 + d$، $J_0 = 4096000$، $J_\mathrm{m} = 8192$، $\epsilon = 0.1$،

$$\gamma_m = 5\left[10^{-2}\mathbf{1}_{\left[1,500+\frac{d}{5}\right]}(m) + 10^{-3}1_{\left(500+\frac{d}{5},1500+\frac{3d}{5}\right]}(m) + 10^{-4}\mathbf{1}_{\left(1500+\frac{3d}{5},\infty\right)}(m)\right]$$

، $\xi_i = \xi_1$، $\mu(x) = (r-\delta)x$، $\sigma(x) = \beta diag(x_1, \ldots, x_d)$،

$$\chi_n^{m-1,j,(i)} = exp\left([r-\delta-0.5\beta^2]t_n + \beta W_{t_n}^{m-1,j,(i)}\right)\xi_i$$

و

$$g(s,x) = e^{-rs} max\{max\{x_1, \ldots, x_d\} - K, 0\}$$

در این مثال به جای ۱۰ بار، ۱ بار قیمت را به دست آورده‌ایم.

مثال ٤.٤.١.٢

در فرمول‌بندی چهار چوب کلی، فرض کنید $r = 0.05$ ، $\delta = 0.1$، $\beta = 0.2$، $K = 100$، $T = 3$، $N = 9$

، $J_0 = 2^{20}$، $J_\mathrm{m} = 1024$، $\epsilon = 10^{-8}$،

$$\gamma_m = \left[10^{-2}\mathbf{1}_{[1,2000]}(m) + 10^{-3}1_{(2000,4000]}(m) + 10^{-4}\mathbf{1}_{(4000,\infty)}(m)\right]$$

، $\xi_1 = 100$، $\mu(x) = (r-\delta)x$، $\sigma(x) = \beta diag(x_1, \ldots, x_d)$،

$$\chi_n^{m-1,j,(i)} = exp\left([r-\delta-0.5\beta^2]t_n + \beta W_{t_n}^{m-1,j,(i)}\right)\xi_i$$

و

$$g(s,x) = e^{-rs} max\{max\{x_1, \ldots, x_d\} - K, 0\}$$

مثال ٤.٤.١.٣

در فرمول‌بندی چهار چوب کلی، فرض کنید $\eta = {}^{30}/_{365}$، $r = 0.05\eta$ ، $\delta = 0.1$، $\beta = 0.4\sqrt{\eta}$،

$K \in \{35, 40, 45\}$، $T = 3$، $N = 9$، $M = 1600$، $J_0 = 4096000$، $J_\mathrm{m} = 8192$، $\epsilon = 0.1$،

$$\gamma_m = 5\left[10^{-2}\mathbf{1}_{[1,400]}(m) + 10^{-3}1_{(400,800]}(m) + 10^{-4}\mathbf{1}_{(800,\infty)}(m)\right]$$

$Q = GG^t$ ،$Q = 0.5\, \mathbf{1}_d \mathbf{1}_d^{\mathrm{t}} + 0.5 I_d$

، $\sigma(x) = \beta diag(x_1, \dots, x_d)\ G$ ،$\mu(x) = \mathrm{r}x$ ،$\xi_i = 40$ ،

$$\chi_n^{m-1,j,(i)} = exp\big([r - \delta - 0.5\beta^2]t_n + \beta\langle \mathrm{G}_{\cdot,\mathrm{i}}, W_{t_n}^{m-1,j}\rangle\big)\xi_i$$

و

$$g(s,x) = e^{-rs} max\{max\{x_1, \dots, x_d\} - K, 0\}$$

در این مثال به جای ۱۰ بار، ۱ بار قیمت را به دست آورده‌ایم.

مثال ٤.٤.٢

فرض کنید

$$G = \begin{pmatrix} 0.3024 & 0.1354 & 0.0722 & 0.1367 & 0.1641 \\ 0.1354 & 0.2270 & 0.0613 & 0.1264 & 0.1610 \\ 0.0722 & 0.0613 & 0.0717 & 0.0884 & 0.0699 \\ 0.1367 & 0.1264 & 0.0884 & 0.2937 & 0.1394 \\ 0.1641 & 0.1610 & 0.0699 & 0.1394 & 0.2535 \end{pmatrix}$$

در فرمول‌بندی چهار چوب کلی، فرض کنید $r = 0.05$ ، $K_2 = 90$ ،$K_1 = 75$ ،$K_3 = 110$ ،$K_4 = 125$ ،$T = 1$، $N = 48$ ،$M = 750$ ،$J_0 = 4096000$ ،$J_{\mathrm{m}} = 8192$ ،$\epsilon = 10^{-8}$،

$\sigma(x) = \beta diag(x_1, \dots, x_d)\ G$ ،$\mu(x) = \mathrm{r}x$ ،$\xi_i = 100$،

$$\gamma_m = 5\big[10^{-2}\mathbf{1}_{[1,250]}(m) + 10^{-3}1_{(250,500]}(m) + 10^{-4}\mathbf{1}_{(500,\infty)}(m)\big]$$

و

$$\chi_n^{m-1,j,(i)} = exp\big(\big[r - 0.5||\mathrm{G}_{\cdot,\mathrm{i}}||^2\big]t_n + \beta\langle \mathrm{G}_{\cdot,\mathrm{i}}, W_{t_n}^{m-1,j}\rangle\big)\xi_i$$

و

$$g(s,x) = -e^{-rs}max\{K_1 - d^{-1}\sum_{k=1}^{d} x_k\,, 0\} + e^{-rs}max\{K_2 - d^{-1}\sum_{k=1}^{d} x_k\,, 0\} + e^{-rs}max\{K_3 - d^{-1}\sum_{k=1}^{d} x_k\,, 0\} - e^{-rs}max\{K_4 - d^{-1}\sum_{k=1}^{d} x_k\,, 0\}$$

مثال ٤.٤.٣

در فرمول‌بندی چهار چوب کلی، فرض کنید $r = 0.05$ ، $\delta \in \{0, 0.1\}$ ،$K = 100$ ،$L = 10$ ،$T = 1$ ،$N = 9$، $M = 1200$ ،$J_0 = 4096000$ ،$J_{\mathrm{m}} = 8192$ ،$\epsilon = 10^{-8}$ ،$d = 5$،

$$\gamma_m = 5\big[10^{-2}\mathbf{1}_{[1,400]}(m) + 10^{-3}1_{(400,800]}(m) + 10^{-4}\mathbf{1}_{(800,\infty)}(m)\big]$$

$\xi_i = 100$ ، $\mu(x) = (r-\delta)x$ ،$\beta: [0,T] \times \mathbb{R} \to \mathbb{R}$ و $\sigma: [0,T] \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ توابعی باشند که برای هر $t \in [0,T]$ و $x = (x_1, \dots, x_d) \in \mathbb{R}^d$ در روابط زیر صدق کنند:

$$\beta(t,x_1) = 0.6e^{-0.05\sqrt{t}}\left(1.2 - e^{-0.1t-0.001(exp(rt)x_1-\xi_1)^2}\right)x_1$$

$$\sigma(t,x) \ = \ diag(\beta(t,x\_1, \dots, \beta(t,x\_d))$$

و$S = \left(S^{(1)}, \dots, S^{(d)}\right): [0,T] \times \Omega \to \mathbb{R}^d$ فرآیند تصادفی پیوسته‌ای باشد که در برای هر $t \ \in [0,T]$ تقریباً مطمئناً داشته باشیم:

$$S_t = \xi + \int_0^t \mu(S_s)ds + \int_0^t \sigma(s,S_s)dW_s^{0,1}$$

فرآیند تصادفی $\mathcal{Y}^{m,j} = \left(\mathcal{Y}^{m,j,(1)}, \dots, \mathcal{Y}^{m,j,(d)}\right): [0,T] \times \Omega \to \mathbb{R}^d$ را برای هر $m \in \mathbb{N}_0$ ،$j \ \in \mathbb{N}$، $l \ \in \{0,1,\dots,L-1\}$، $t \in \left[\frac{lT}{L}, \frac{(l+1)T}{L}\right]$ ،$i \ \in \{1,\dots,d\}$ داشته باشیم $\mathcal{Y}_0^{m,j,(i)} = log(\xi_i)$ و

$$\mathcal{Y}_t^{m,j,(i)} = \mathcal{Y}_{\frac{lT}{L}}^{m,j,(i)} + \left(t - \frac{lT}{L}\right)\left(r - \delta - 0.5\left[\beta\left(\frac{lT}{L}, exp\left(\mathcal{Y}_{\frac{lT}{L}}^{m,j,(i)}\right)\right)\right]^2\right) + \left(\frac{tL}{T} - l\right)\beta\left(\frac{lT}{L}, exp\left(\mathcal{Y}_{\frac{lT}{L}}^{m,j,(i)}\right)\right)\left(W_{\frac{(l+1)T}{L}}^{m,j,(i)} - W_{\frac{lT}{L}}^{m,j,(i)}\right)$$

و تابع عایدی برابر با

$$g(s,x) = e^{-rs}max\{K - d^{-1}\left[\sum_{i=1}^{d} exp(x_i)\right], 0\}$$

مثال ٤.٤.٤

در فرمول‌بندی چهار چوب کلی، فرض کنید $r \ = \ 0.0004$ ، $\beta \ = \ 0.02$ ،$\mathcal{W}^{m,j}: [0,\infty) \times \Omega \to \mathbb{R}$ برای $j \ \in \mathbb{N}$ و $m \in \mathbb{N}_0$ حرکات براونی استاندارد و مستقل باشند. $S^{m,j}: [-100,\infty) \to \mathbb{R}$ و $\mathcal{Y}^{m,j}: \mathbb{N}_0 \times \Omega \to \mathbb{R}^{100}$ برای $j \ \in \mathbb{N}$ و $m \in \mathbb{N}_0$ و $t \in [-100,\infty)$، فرآیندهای تصادفی باشند که در روابط زیر صدق می‌کنند:

$$S_t^{m,j} = exp\left([r - 0.5\beta^2](t+100) + \beta\mathcal{W}_{t+100}^{m,j}\right)\xi_1$$

$$\mathcal{Y}_n^{m,j} = \left(\frac{S_{n-99}^{m,j}}{S_{n-100}^{m,j}}, \frac{S_{n-98}^{m,j}}{S_{n-100}^{m,j}}, \dots, \frac{S_n^{m,j}}{S_{n-100}^{m,j}}\right)$$

فرض کنید $d \ = \ 100$ ،$N = T$ ،$\chi_n^{m-1,j} = \mathcal{Y}_n^{m-1,j}$ ،$J_0 = 4096000$ ،$\epsilon = 10^{-8}$

$$\mathrm{M} = \left[1200\ \mathbf{1}_{[1,150]}(\mathrm{T}) + 1500\ 1_{(150,250]}(\mathrm{T}) + 3000\ \mathbf{1}_{(250,\infty)}(\mathrm{T})\right]$$

$$\gamma_m = 5\left[10^{-2}\mathbf{1}_{[1,\mathrm{M}/_3]}(m) + 10^{-3}1_{(\mathrm{M}/_3, 2\mathrm{M}/_3]}(m) + 10^{-4}\mathbf{1}_{(2\mathrm{M}/_3,\infty)}(m)\right]$$

$$J_{\mathrm{m}} = 8192\ \mathbf{1}_{[1,50]}(d) + 4096\ \mathbf{1}_{(50,100]}(d) + 2048\ \mathbf{1}_{(100,\infty)}(d)$$

و

$$g(s,x) = e^{-rs} x_{100}$$

در این مثال، سایت Colab خطای محدودیت فضای GPU را داد و شبیه‌سازی تنها برای $T = 100, 150$ قابل اجرا بود.