

Министерство образования и науки Российской Федерации  
Московский физико-технический институт  
(национальный исследовательский университет)

Физтех-школа прикладной математики и информатики  
Кафедра банковских информационных технологий

Выпускная квалификационная работа бакалавра по направлению 03.03.01  
«Прикладные математика и физика»

## Оценка стоимости азиатских опционов с помощью метода Монте Карло и машинного обучения

Студент Б05-812 группы  
Яскевич А. С.

Научный руководитель  
Куликов А. В.

Москва  
2022



# Содержание

<b>1. Теоретическая часть</b>	<b>1</b>
1.1. Введение . . . . .	1
1.2. Основные характеристики и виды опционов . . . . .	3
1.2.1. Европейские опционы . . . . .	3
1.2.2. Американские опционы . . . . .	4
1.2.3. Азиатские опционы . . . . .	5
1.3. Обзор литературы . . . . .	6
1.3.1. Application of machine learning: an analysis of asian options using neural network [1] . . . . .	6
1.3.2. Accelerating Python for Exotic Option Pricing [2] . . . . .	6
1.3.3. Universal function approximation by deep neural nets with bounded width and ReLU activations [3] . . . . .	8
1.3.4. Incorporating Functional Knowledge in Neural Networks [4] . . . . .	8
1.4. Цель и структура работы . . . . .	10
1.5. Доказательство свойств опционов . . . . .	10
1.5.1. Зависимость цены от отношения $\frac{S_0}{K}$ . . . . .	10
1.5.2. Put Call Parity . . . . .	11
1.5.3. Греки . . . . .	11
1.5.4. Выпуклость цены относительно страйка . . . . .	12
1.5.5. Убывание цены Call-опциона при возрастании страйка . . . . .	12
1.6. Монте Карло . . . . .	13
1.6.1. Общая идея . . . . .	13
1.6.2. Доверительные интервалы . . . . .	14
1.6.3. Antithetic variates . . . . .	15
1.7. Архитектуры нейронных сетей . . . . .	16
1.7.1. Архитектура, гарантирующая положительность цены и ее воз- растание по $\frac{S_0}{K}$ . . . . .	16
1.7.2. Архитектура, гарантирующая выпуклость цены относительно $\frac{S_0}{K}$ . . . . .	17
<b>2. Практическая часть</b>	<b>19</b>
2.1. Реализация метода Монте Карло . . . . .	19
2.2. Реализация и обучение нейронных сетей . . . . .	21

2.2.1.	Архитектура, гарантирующая положительность и возрастание цены . . . . .	21
2.2.2.	Архитектура, гарантирующая выпуклость цены относительно $\frac{S_0}{K}$	24
2.3.	Сравнение результатов . . . . .	26
2.4.	Выводы . . . . .	27





# Глава 1

## Теоретическая часть

### 1.1 Введение

В последние годы получили большое распространение так называемые азиатские опционы [5] – производные финансовые инструменты, позволяющие участникам рынка защищаться от ценовых рисков. В отличие от европейских или американских опционов, цена экспирации по которым определяется ценой базового актива строго в дату экспирации, для азиатских опционов она вычисляется как среднее цены базового актива за определенный период. Это позволяет получить более стабильный финансовый результат, который менее подвержен воздействию резкого изменения цен.

Наиболее часто азиатские опционы применяются на товарных рынках, а частности – на рынке нефти. Из-за того, что нефть необходимо физически доставлять от продавца покупателю, а также осуществлять хранение, иногда возникают локальные избытки или недостатки. Это может повлечь за собой резкое изменение котировок, причем как в сторону их повышения, так и понижения, включая переход к отрицательным значениям. Как правило, эти изменения носят краткосрочный характер, в связи с чем усреднение цены базового актива позволяет сделать опцион менее рискованным как для продавцов, так и для покупателей.

В качестве примера можно привести программу хеджирования нефтяных доходов Мексики [6], принятую в 2001 году. Основной ее задачей является снижение риска падения доходов бюджета в каждом конкретном году. Министерство финансов Мексики закладывает в бюджет экспортную цену нефти, рассчитываемую как взвешенная сумма исторических цен и цен фьючерсов, а затем покупает азиатские Put-опционы со страйком, близким к этой цене. По данным МВФ [6], таким образом хеджируется 29% добываемой в Мексике нефти. Затраты и выплаты по программе хеджирования, а также страйк в различные годы представлены на Рис. 1.

Кроме описанных выше преимуществ азиатских опционов, существует и значительный недостаток: их цены невозможно получить аналитически, в отличие от европейских и американских, для которых применима формула Блэка-Шоулза. Поэтому

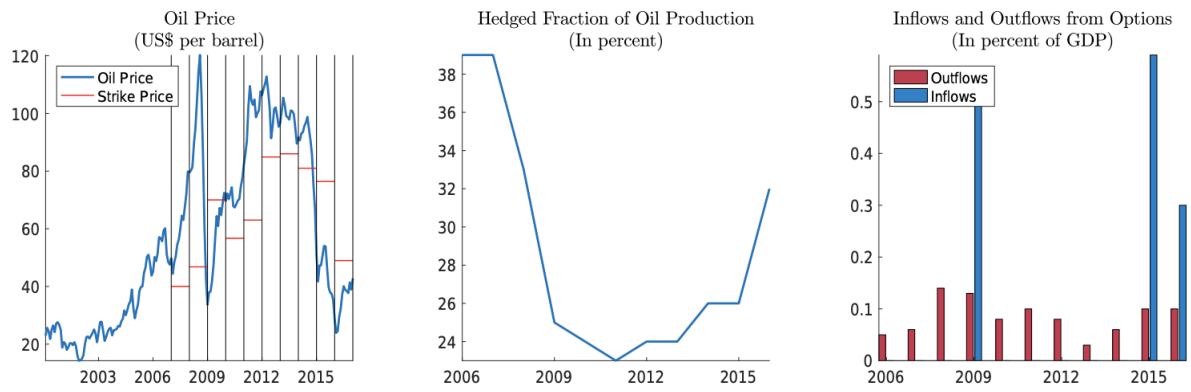


Рис. 1. Программа хеджирования Мексики

применяется метод Монте Карло, который будет разобран далее, но скорость его работы не позволяет предлагать лучшие цены на рынке. Добиться значительно лучших результатов можно с помощью методов машинного обучения.



## 1.2 Основные характеристики и виды опционов

### 1.2.1 Европейские опционы

Европейский опцион – это производный финансовый инструмент, который дает его владельцу право купить (продать) базовый актив в заранее установленную дату (дата экспирации) по заранее определенной цене (страйк). По типу их разделяют на Call-опционы, дающие право на покупку, и Put-опционы, дающие право на продажу. Для цены европейского Call-опциона существует аналитическое выражение, называемое формулой Блэка-Шоулза [5].

$$C = N(d_1)S_t - N(d_2)Ke^{-rt},$$

где

$$d_1 = \frac{\ln \frac{S_t}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}, \quad d_2 = d_1 - \sigma\sqrt{t}.$$

$N(x)$  – функция нормального распределения

$S_t$  – цена базового актива

$K$  – страйк

$r$  – безрисковая ставка

$t$  – время до экспирации

$\sigma$  – волатильность

Цену Put-опциона можно получить из соотношения, называемого Put-Call Parity (паритет опционов пут и колл), которое выглядит следующим образом.

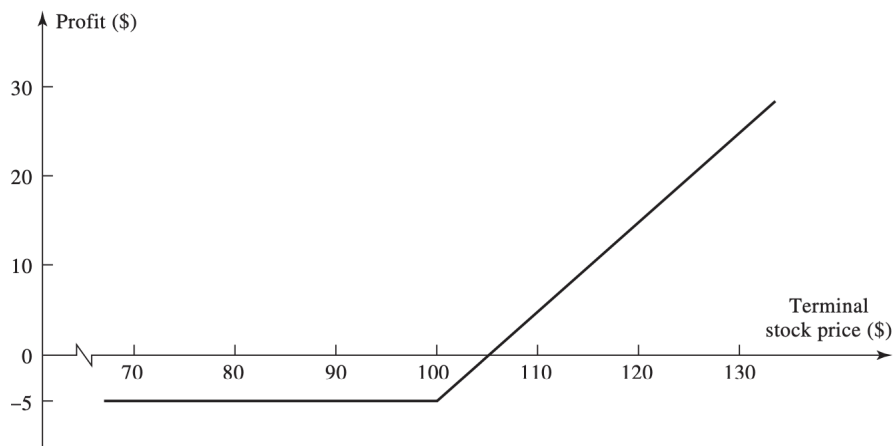
$$C(t) - P(t) = S(t) - e^{-r(T-t)}K,$$

где  $C(t)$  – цена Call-опциона,  $P(t)$  – цена Put-опциона,  $S(t)$  – цена базового актива,  $K$  – страйк.

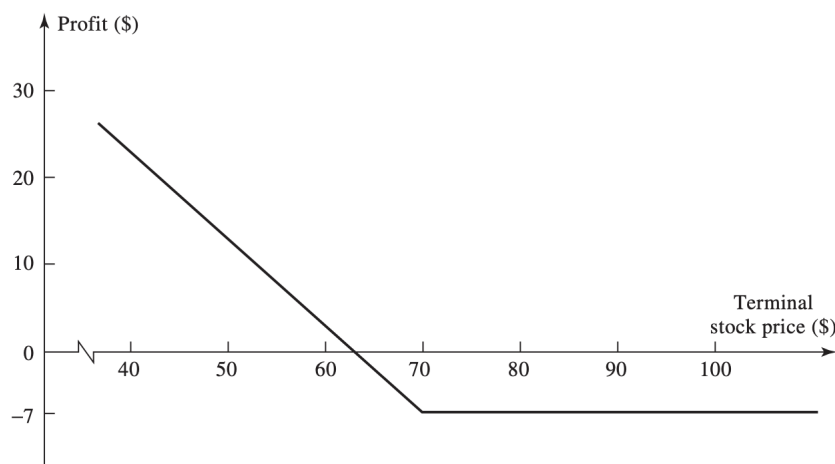
Формула для выплаты по опционам:  $(S_t - K)^+$  для Call-опциона и  $(K - S_t)^+$  для Put-опциона.

Прибыль от использования опционов изображена на следующих графиках.

**Figure 10.1** Profit from buying a European call option on one share of a stock. Option price = \$5; strike price = \$100.



**Figure 10.2** Profit from buying a European put option on one share of a stock. Option price = \$7; strike price = \$70.



### 1.2.2 Американские опционы

Американский опцион – это производный финансовый инструмент, аналогичный европейскому опциону за тем исключением, что предъявить его к исполнению можно либо в дату экспирации, либо в любое время до нее. Как следствие, покупатель американского опциона получает больше прав, чем покупатель европейского.

В общем случае для цены американского опциона нет аналитического выражения, но в случае Call-опциона на базовый актив, который не выплачивает дивидендов, его цена совпадает с ценой европейского Call-опциона.

Формула для выплаты по опционам:  $(S_t - K)^+$  для Call-опциона и  $(K - S_t)^+$  для Put-опциона.

Европейские и американские опционы наиболее популярны среди всех остальных опционов и торгуются на многих биржах, а также на внебиржевом рынке.

### 1.2.3 Азиатские опционы

Азиатский опцион – это опцион, выплата по которому определяется ценой базового актива не в дату экспирации, а на основе усреднения за заранее оговоренный промежуток времени. Усреднение может быть арифметическим или геометрическим. В результате снижается волатильность, а следовательно, и цена опциона.

Азиатские опционы относятся к классу экзотических и применяются реже, чем европейские или американские. Они торгуются преимущественно на внебиржевом рынке, однако некоторые контракты (например, на хлопок или алюминий) торгуются на организованных торгах, в частности – на Лондонской бирже металлов [11] (LME). Они являются расчетными, а не поставочными, предполагают арифметическое усреднение цены и называются TAPOs (Traded Average Price Options, торгуемые опционы средней цены)

Выплаты по Call и Put-опционам задаются формулами  $(A(0, T) - K)^+$  и  $(K - A(0, T))^+$  соответственно. Для опционов с арифметическим усреднением

$$A(0, T) = \frac{1}{T} \sum_{i=1}^T S_i.$$

Стоит еще раз заметить, что для данного вида экзотических опционов не существует аналитического выражения их цены. Для опционов с геометрическим усреднением:

$$A(0, T) = \exp\left(\frac{1}{N} \int_0^T \ln S(t) dt\right)$$

Для них есть возможность получить аналитическое выражение для цены, так как геометрическое среднее и процесс цены базового актива обладают одним важным общим свойством – логнормальным распределением. Приведем здесь формулу для нахождения цены. Введем обозначения:

$$b = \frac{1}{2} \left( r - \frac{1}{2} \sigma_G^2 \right), \quad \sigma_G = \frac{\sigma}{\sqrt{3}}, \quad d_1 = \frac{\log \frac{S_0}{K} + (b + \frac{1}{2} \sigma_G^2) T}{\sigma_G \sqrt{T}}$$

$$d_2 = d_1 - \sigma_G \sqrt{T}$$

Тогда цена запишется как

$$C_G = S_0 e^{(b-r)T} \Phi(d_1) - K e^{-rT} \Phi(d_2)$$

## 1.3 Обзор литературы

### 1.3.1 *Application of machine learning: an analysis of asian options using neural network [1]*

В статье рассматриваются основные модели, позволяющие приближенно аналитически найти цены азиатских опционов, и их основные недостатки. В частности, подробно рассматриваются две популярные на сегодняшний день модели: модель Кемна-Ворста [8] и приближение Леви [9], которые, однако, не всегда дают достаточную точность и надежность.

В модели Кемна-Ворста предполагается, что цена базового актива усредняется геометрически, в то время как на самом деле чаще всего используется среднее арифметическое. В связи с этим, в этой модели опционы всегда являются недооцененными, что прямо следует из неравенства между средним арифметическим и геометрическим.

Что касается приближения Леви, то оно приближает распределение среднего арифметического через логнормальное распределение, имеющее такие же первый и второй моменты. Это позволяет использовать аналог формулы Блэка-Шоулза, но при больших значениях волатильности и/или времени до экспирации точность такого приближения значительно снижается.

Основная идея статьи заключается в том, чтобы с помощью машинного обучения улучшать результаты, получаемые с помощью аналитических приближений цен. Так, автор использует нейронную сеть для получения implied volatility (прогнозируемая волатильность) на основе текущих параметров, а затем использует ее в аналитических моделях. В качестве бенчмарка используется метод Монте Карло с применением Antithetic Variates (антитетические переменные) для понижения дисперсии получаемых цен.

В результате автору удалось добиться улучшения расчета цены для тех опционов, на которых модель Леви работала плохо, а именно – для опционов, для которых  $\sigma\sqrt{t} > 0.2$ . Среднеквадратичное отклонение между ценами Монте Карло и ценами, получаемыми через приближение Леви, удалось снизить на 4 порядка.

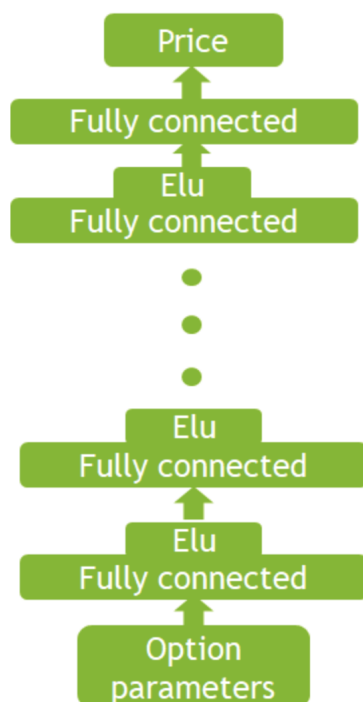
### 1.3.2 *Accelerating Python for Exotic Option Pricing [2]*

В статье поднимается проблема увеличения эффективности вычислений для получения большей прибыли при торговле, а также нахождения баланса между сложностью моделей и временем вычисления. Большая часть исследовательского кода написана на Python, который может работать слишком медленно для промышленных задач. В данной статье описываются способы ускорить работу Python в задаче ценообразования экзотических опционов.

В первой части статьи описываются различные способы реализации метода Монте Карло. Приведем их краткое описание:

1. CUDA C/C++. Это очень детальный код на C++, в котором происходит выделение памяти на GPU, использование библиотеки cuRand для генерации случайных чисел и, соответственно, расчета траекторий. Затем происходит параллельное вычисление цен опционов как дисконтированного математического ожидания выплаты по опциону, после чего результаты агрегируются, и память деаллоцируется. По данным из статьи, расчет цены на 8.192 млн траекторий из 365 шагов занял 26.6 мс. Далее в этой статье все эксперименты проводятся на этих же данных.
2. Numba на CPU. Numba – это библиотека, позволяющая выполнять JIT-компиляцию кода на Python. Был использован компилятор fastmath, который дополнительно ускоряет вычисления, но с меньшими гарантиями по точности. Также есть многопоточный режим. Расчет цен занял 41.6 с и 2.34 с для однопоточной и многопоточной версий соответственно.
3. Numba на GPU. Также можно производить вычисления на GPU, используя соответствующий декоратор. Авторам статьи удалось добиться снижения времени работы до 65 мс.
4. Библиотека CuPy на GPU. Она позволяет использовать код на C++ для вычислений на GPU, причем подключиться к ней можно непосредственно из Python. Время работы составило 29 мс, что достаточно близко к результату из пункта 1.

Во второй части статьи автор использует нейронную сеть для решения той же задачи. Ее архитектура представлена на следующей схеме.



В качестве функции потерь был использован MSELoss (среднеквадратичная функция потерь), а в качестве оптимизатора – Adam. В режиме inference (предсказание) были рассчитаны цены опционов, что заняло 0.8 мс, или в 32 раза быстрее, чем метод, основанный на CUDA. Также на основании градиентов нейронной сети были рассчитаны греки опциона.

### 1.3.3 Universal function approximation by deep neural nets with bounded width and ReLU activations [3]

В данной статье доказываются различные утверждения относительно того, какой ширины внутренних слоев сети достаточно для приближения функций с различными свойствами. Как следует из названия, в статье рассматриваются именно нейронные сети с функцией активации ReLU (Rectified Linear Unit, выпрямленная линейная функция).

Можно выделить следующие результаты. Рассмотрим  $d \geq 1$  и  $f : [0, 1]^d \rightarrow R_+$ , такую что  $\|f\|_{C_0} = 1$ . Тогда есть три случая, в каждом из которых автором получены полезные результаты.

1. Если  $f$  непрерывна. Тогда существует последовательность нейронных сетей  $N_k$  с функциями активации ReLU, размерностью пространства признаков  $d$ , шириной внутреннего слоя  $d + 2$  и размерностью 1 на выходе, что

$$\lim_{k \rightarrow \infty} \|f - f_{N_k}\|_{C_0} = 0.$$

Если же увеличить разрешенную ширину внутреннего слоя до  $d + 3$ , то можно получить оценку на достаточную глубину сети, для которой будет справедливо неравенство  $\|f - f_{N_\varepsilon}\|_{C_0} \leq \varepsilon$ :

$$\text{depth}(N_\varepsilon) = \frac{2 \cdot d!}{\omega_f(\varepsilon)^d}$$

2. Если  $f$  выпукла. Тогда существует последовательность нейронных сетей, аналогичная пункту 1, за тем исключением, что ширина внутреннего слоя может быть  $d + 1$ . Кроме того, существует  $C > 0$ , такое что для липшицевых функций с константой Липшица  $L$  может быть выбрана такая последовательность  $\{f_{N_k}\}$ , что

$$\text{depth}(N_k) = k + 1; \quad \|f - f_{N_k}\|_{C^0} \leq CLd^{\frac{3}{2}}k^{-\frac{2}{d}}$$

### 1.3.4 Incorporating Functional Knowledge in Neural Networks [4]

Эта статья важна с точки зрения выбора архитектуры нейронной сети для предсказания цен опционов. Известно, что они обладают следующими свойствами: моно-

тонное возрастание цены Call-опциона относительно параметра "moneyness" (денежность)  $= \frac{S_0}{K}$  и времени до экспирации, а также выпуклость вниз цены относительно параметра "moneyness" который вычисляется как отношение цены базового актива к страйку.

Изначально авторы берут модель, состоящую из двух полносвязных слоев и сигмоидой между ними. Далее они переходят к использованию разных функций активации для разных параметров. Так, для параметров, относительно которых цена опциона выпукла, используется выпуклая функция активации softplus:  $\zeta(x) = \ln(1 + e^x)$ , а для остальных параметров используется сигмоида  $\sigma(x) = \frac{1}{1+e^{-x}}$ . Кроме того, в данной архитектуре веса обязаны быть положительными, что достигается с помощью перехода от самих весов к их экспонентам. Архитектура принимает такой вид:

$$N_{++} = \left\{ f(x) = e^{w_0} + \sum_{i=1}^H e^{w_i} \left( \prod_{j=1}^c \zeta(b_{ij} + e^{\nu_{ij}} x_j) \right) \left( \prod_{j=c+1}^n h(b_{ij} + e^{\nu_{ij}} x_j) \right) \right\}$$

Mean Squared Error Results on Call Option Pricing ( $\times 10^{-4}$ )

Hidden Units	UPSD				CPSD			
	Train	Valid	Test1	Test2	Train	Valid	Test1	Test2
1	2.27	2.15	2.35	3.27	2.28	2.14	2.37	3.51
2	1.61	1.58	1.58	14.24	2.28	2.13	2.37	3.48
3	1.51	1.53	1.38	18.16	2.28	2.13	2.36	3.48
4	1.46	1.51	1.29	20.14	1.84	1.54	<b>1.97</b>	<b>4.19</b>
5	1.57	1.57	1.46	10.03	1.83	1.56	1.95	4.18
6	1.51	1.53	1.35	22.47	1.85	1.57	1.97	4.09
7	1.62	1.67	1.46	7.78	1.86	1.55	2.00	4.10
8	1.55	1.54	1.44	11.58	1.84	1.55	1.96	4.25
9	1.46	1.47	<b>1.31</b>	<b>26.13</b>	1.87	1.60	1.97	4.12

Hidden Units	USSD				CSSD			
	Train	Valid	Test1	Test2	Train	Valid	Test1	Test2
1	1.83	1.59	1.93	4.10	2.30	2.19	2.36	3.43
2	1.42	1.45	<b>1.26</b>	<b>25.00</b>	2.29	2.19	2.34	3.39
3	1.45	1.46	1.32	35.00	1.84	1.58	1.95	4.11
4	1.56	1.69	1.33	21.80	1.85	1.56	1.99	4.09
5	1.60	1.69	1.42	10.11	1.85	1.52	<b>2.00</b>	<b>4.21</b>
6	1.57	1.66	1.39	14.99	1.86	1.54	2.00	4.12
7	1.61	1.67	1.48	8.00	1.86	1.60	1.98	3.94
8	1.64	1.72	1.48	7.89	1.85	1.54	1.98	4.25
9	1.65	1.70	1.52	6.16	1.84	1.54	1.97	4.25

Рис. 2.

Далее авторами было проведено сравнение различных архитектур по метрике MSE. Была подтверждена гипотеза о том, что архитектуры, гарантирующие поло-

жительность, возрастание и выпуклость цены, обладают лучшей обобщающей способностью по сравнению с другими архитектурами. Результаты экспериментов из статьи можно увидеть в таблице на рис. 2. Архитектуре со всеми ограничениями соответствуют данные по CPSD.

## 1.4 Цель и структура работы

Цель работы заключается в том, чтобы принять или отвергнуть гипотезу о том, что использование методов машинного обучения позволяет вычислять цены азиатских опционов, а также их производные, с большей скоростью, чем использование только метода Монте Карло. Работа будет состоять из следующих частей:

1. Реализация расчета цен опционов с помощью метода Монте Карло, нахождение доверительных интервалов
2. Применение методов снижения дисперсии для метода Монте Карло и улучшение результата
3. Обучение нейронной сети для определения цен
4. Обучение сети, гарантирующей экономически важные свойства цен опционов: неотрицательность, убывание при возрастании страйка, а также выпуклость относительно него.
5. Реализация расчета греков (их определения и смысл даны ниже)
6. Сравнение результатов Монте Карло и машинного обучения

## 1.5 Доказательство свойств опционов

### 1.5.1 Зависимость цены от отношения $\frac{S_0}{K}$

Докажем, что цена опциона зависит не от изначальной цены базового актива и страйка по отдельности, а от их частного. Это позволит снизить размерность пространства признаков на 1.

Запишем формулу для стоимости опциона как дисконтированное матожидание выплат в риск-нейтральной мере.

$$V = e^{-rT} \mathbb{E} (A(0, T) - K)^+$$

Разделим обе части уравнения на  $K$ . Тогда:

$$\frac{V}{K} = e^{-rT} \mathbb{E} \left( \frac{A(0, T)}{K} - 1 \right)^+$$



Учитывая, что  $A(0, T) = \frac{1}{T} \sum_{i=1}^T S_i$  при использовании среднего арифметического и  $A(0, T) = \exp\left(\frac{1}{N} \int_0^T \ln S(t) dt\right)$  для среднего геометрического, можно заключить, что цена базового актива и страйк входят в формулу только в виде частного. Так мы смогли снизить количество параметров на 1. Осталось только домножить полученную цену на страйк.

### 1.5.2 Put Call Parity

Чтобы не вычислять отдельно цены для азиатских Put и Call опционов, покажем, что они связаны между собой определенным соотношением, аналогичным соотношению для европейских и американских опционов. Оно называется Put-Call Parity (паритет опционов пут и колл). Далее обозначаем цену Call-опциона как  $C(t)$ , а цену Put-опциона – как  $P(t)$ . Справедлива формула:

$$C(T) - P(T) = (A(0, T) - K)^+ - (K - A(0, T))^+ = A(0, T) - K$$

В момент времени  $t \in [0, T]$  нужно дисконтировать страйк:

$$C(t) - P(t) = A(0, t) - e^{-r(T-t)}K$$

Видим, что, зная цену Call-опциона в момент времени  $t$ , по формуле выше сможем вычислить формулу Put-опциона, что избавляет нас от необходимости учитывать этот параметр (Put/Call) в дальнейшем.

### 1.5.3 Греки

В задачах, связанных с опционами, часто бывают нужны не только их цены, но и их производные по различным параметрам. Наиболее популярные из них называются греками (the Greeks) и записываются следующим образом:

1. Дельта:  $\Delta = \frac{\partial V}{\partial S}$ . Показывает, как изменяется цена опциона при изменении цены базового актива. Используется при так называемом дельта-хеджировании, суть которого заключается в том, что если покупать базовый актив в объеме, равном значению дельты со знаком минус (при положительной дельте – продавать), то получится полностью захеджировать портфель.
2. Вега:  $\nu = \frac{\partial V}{\partial \sigma}$ . Показывает чувствительность цены опциона к волатильности базового актива. При прочих равных условиях, цена опциона растет при росте волатильности. Для европейских опционов это прямо следует из формулы Блэка-Шоулза.
3. Тета:  $\Theta = -\frac{\partial V}{\partial \tau}$ . Показывает, как изменяется цена с течением времени при прочих равных условиях. Если разделить цену опциона на внутреннюю (intrinsic)

и временную (time), то тета изменяет именно изменение временной стоимости с течением времени (time decay).

4. Рho:  $\rho = \frac{\partial V}{\partial r}$ . Показывает чувствительность цены опциона к безрисковой ставке. При ее возрастании обычно цена опциона падает, однако эта зависимость наиболее слабая по сравнению с другими греками.
5. Гамма:  $\Gamma = \frac{\partial^2 V}{\partial S^2}$ . Вторая производная цены опциона по цене базового актива. У Call-опционов гамма положительная (цена является выпуклой функцией), у Put-опционов – отрицательная.

Также используются и другие производные цены, однако наиболее часто применяются описанные выше пять видов.

### 1.5.4 Выпуклость цены относительно страйка

Пусть  $K$  – страйк,  $V(K)$  – цена опциона. Покажем ее выпуклость, т.е. то, что для любого  $\delta > 0$   $V(K - \delta) + V(K + \delta) > 2V(K)$ . Предположим, что это неверно, и существует такой контракт и такое  $\delta > 0$ , что

$$V(K - \delta) + V(K + \delta) \leq 2V(K)$$

Тогда рассмотрим следующую стратегию: покупаем опционы со страйком  $K - \delta$  и  $K + \delta$ , продаем опцион со страйком  $K$ . Мы можем это сделать из неравенства выше, а оставшиеся деньги (при наличии) разместить в банке. Во время экспирации выплата составит:

$$P = (S - (K - \delta))^+ + (S - (K + \delta))^+ - 2(S - K)^+$$

Рассмотрим следующие варианты:

1.  $S \leq K - \delta$ . Тогда  $P = 0$ .
2.  $K - \delta < S < K$ . Тогда  $P = S - (K - \delta) > 0$
3.  $K \leq S < K + \delta$ . Тогда  $P = S - K + \delta - 2(S - K) = K + \delta - S > 0$
4.  $S \geq K + \delta$ . Тогда  $P = S - K + \delta + S - K - \delta - 2(S - K) = 0$

Таким образом, существует стратегия, при которой есть ненулевая вероятность заработать и нет возможности потерять деньги. Так как предполагается, что арбитраж невозможен, то цена действительно выпукла относительно страйка.

### 1.5.5 Убывание цены Call-опциона при возрастании страйка

Аналогично, пусть  $K$  – страйк,  $V(K)$  – цена опциона. Покажем, что для любых  $K$  и  $\delta > 0$   $V(K + \delta) < V(K)$ .

Пусть для некоторых  $K$  и  $\delta$   $V(K+\delta) \geq V(K)$ . Тогда продадим опцион со страйком  $K + \delta$ , купим опцион со страйком  $K$ , который не дороже в силу неравенства. Во время экспирации мы получим выплату  $P = (S - K)^+ - (S - K - \delta)^+$ . По аналогии с предыдущим пунктом, получаем возможность арбитража, а значит предположение неверно, и цена убывает при возрастании страйка. Так как moneyness определяется как  $\frac{S_0}{K}$ , то цена возрастает относительно этого параметра.

## 1.6 Монте Карло

### 1.6.1 Общая идея

С помощью метода Монте Карло можно получить цену любого опциона с помощью следующих шагов:

1. Сгенерировать случайную траекторию движения цены базового актива в риск-нейтральной мере
2. Вычислить выплату по производному инструменту при условии, что поведение базового актива соответствует траектории из пункта 1
3. Повторить пункты 1 и 2, чтобы получить множество выплат по производному инструменту в риск-нейтральной мере
4. Рассчитать среднюю выплату по данным из пункта 3, чтобы оценить матожидание выплаты в риск-нейтральной мере
5. Дисконтировать полученную оценку для матожидания выплаты, чтобы получить оценку цены производного инструмента

В основе метода лежит предположение о том, что движение базового актива определяется следующим процессом:

$$dS = \mu S dt + \sigma S dz,$$

где  $dz$  – это винеровский процесс [5],  $\mu$  – матожидание прибыли в риск-нейтральной мере,  $\sigma$  – волатильность. Если по базовому активу не выплачиваются дивиденды, то  $\mu = r$ , где  $r$  – безрисковая ставка. Далее будем считать, что данное условие выполнено.

На практике удобнее использовать  $\ln S$ , чем  $S$ . Из леммы Ито [5] получаем, что  $\ln S$  подчиняется процессу:

$$d\ln S = \left(r - \frac{\sigma^2}{2}\right) dt + \sigma dz$$

Перейдем к конечным разностям:

$$\ln S(t + \Delta t) - \ln S(t) = \left( r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \varepsilon \sqrt{\Delta t}, \quad \varepsilon \sim N(0, 1)$$

Экспоненцируем и получаем следующее выражение:

$$S(t + \Delta t) = S(t) \exp \left[ \left( r - \frac{\sigma^2}{2} \right) \Delta t + \sigma \varepsilon \sqrt{\Delta t} \right]$$

Если  $r$  и  $\sigma$  постоянны, то справедлива следующая запись:

$$S(T) = S(0) \exp \left[ \left( r - \frac{\sigma^2}{2} \right) T + \sigma \varepsilon \sqrt{T} \right]$$

Эта формула используется для генерации траекторий цены базового актива. Пример сгенерированных траекторий можно видеть на графике 3 в практической части.

### 1.6.2 Доверительные интервалы

Точность результата, полученного методом Монте Карло, зависит от числа сгенерированных траекторий. Чтобы оценить точность результата, помимо самого значения рассчитывается также стандартное отклонение. Обозначим найденное среднее значение как  $\mu$ , цены на каждой из  $N$  траекторий  $X_i$ , а стандартное отклонение как  $\omega$ . Вычисление  $\omega$  производится по известной несмещенной оценке для дисперсии:

$$\omega^2 = \frac{\sum_{i=1}^N (X_i - \mu)^2}{N - 1}$$

Пусть  $f$  – истинное значение цены опциона. Тогда по центральной предельной теореме  $\mu - f \sim N(0, \frac{\omega^2}{N})$ . Отсюда находим выражение для доверительного интервала уровня значимости 0.95:

$$\left[ \mu - \frac{1.96\omega}{\sqrt{N}}, \mu + \frac{1.96\omega}{\sqrt{N}} \right].$$

Здесь использован множитель 1.96, так как  $\Phi(1.96) = 0.975$ , а  $\Phi(-1.96) = 0.025$ . Таким образом, истинное значение цены лежит в указанном отрезке в 95% случаев.

В общем случае, когда требуется выразить доверительный интервал уровня доверия  $\alpha$ , множителем вместо 1.96 будет число  $A$ , такое что  $\Phi(A) = 1 - \frac{1-\alpha}{2}$ , где  $\Phi(x)$  – функция распределения стандартного нормального распределения.

Пример рассчитанных цен и доверительных интервалов можно видеть в таблице 1 в практической части.

Ширина доверительного интервала зависит от двух факторов: от количества симуляций и от дисперсии случайных величин  $X_i$  (в нашем случае – цен опционов). Доверительный интервал сужается как корень из числа симуляций, что делает метод Монте Карло недостаточно точным. Но он сужается линейно относительно сред-

неквадратичного отклонения. Таким образом, результат можно улучшить без многократного увеличения ресурсов, если применить методики понижения дисперсии. Одним из таких методов является описанный ниже метод *antithetic variates* (метод антитетических переменных).

### 1.6.3 *Antithetic variates*

Идея метода состоит в следующем. Пусть есть случайная величина  $X$ . Рассмотрим другую случайную величину  $Z$ , такую что  $\mathbb{E}X = \mathbb{E}Z$  и  $cov(X, Z) \leq 0$ . Рассмотрим их сумму  $Y = \frac{X+Z}{2}$ . Тогда  $\mathbb{E}Y = \mathbb{E}X$  и

$$VarY = cov\left(\frac{X+Z}{2}, \frac{X+Z}{2}\right) = \frac{1}{4}(VarX + 2cov(X, Z) + VarZ) \leq \frac{1}{2}VarX$$

Таким образом, дисперсия снизилась, по крайней мере, в два раза. Сложность заключается лишь в том, что матожидание  $X$  неизвестно. Но имеем следующее утверждение: если  $X$  – произвольная случайная величина, и  $f$  – монотонная функция, то  $cov(f(X), f(-X)) \leq 0$

Рассмотрим теперь случайную величину вида  $f(U)$ , где  $U \sim N(0, 1)$ . Из свойств нормального распределения получаем, что  $-U \sim N(0, 1)$ , то есть  $f(U) \sim f(-U)$ . Определим  $Y = \frac{f(U)+f(-U)}{2}$ .  $f$  монотонна, и из предыдущего утверждения  $cov(f(U), f(-U)) \leq 0$ .

Окончательно получаем, что

$$\mathbb{E}\left(\frac{f(U) + f(-U)}{2}\right) = \mathbb{E}f(U)$$

$$Var\left(\frac{f(U) + f(-U)}{2}\right) \leq \frac{1}{2}Varf(U)$$

Таким образом, если в дополнение к стандартной траектории метода Монте Карло генерировать также перевернутую траекторию, то получается снизить дисперсию, а значит и ширину доверительного интервала, как минимум вдвое. Формула для генерации траектории записывается так:

$$S(T) = S(0)exp\left[\left(r - \frac{\sigma^2}{2}\right)T - \sigma\varepsilon\sqrt{T}\right]$$

Заметим, что на рисунке 3 отчетливо видны траектории, сгенерированные с помощью метода *antithetic variates*.

Данные, сгенерированные с использованием данного метода, можно видеть в таблице 2 в практической части.

## 1.7 Архитектуры нейронных сетей

### 1.7.1 Архитектура, гарантирующая положительность цены и ее возрастание по $\frac{S_0}{K}$

Построение архитектур, учитывающих специфические свойства решаемой задачи, позволяет снизить сложность модели и гарантировать выполнение этих свойств во всех случаях, независимо от размера датасета, времени обучения и других обстоятельств. Вместе с этим обычно повышается качество модели, а также снижается время обучения. В этой работе в качестве априорных знаний о цене Call-опциона мы будем принимать то, что цена всегда неотрицательна, монотонно возрастает по параметру "moneyness" (денежность)  $= \frac{S_0}{K}$  и времени до экспирации, а также выпукла относительно параметра moneyness.

Математически это означает следующее. Если выразить цену как функцию денежности и времени до экспирации  $f(x_1, x_2)$ , то получим следующие свойства:

$$f \geq 0, \quad \frac{\partial f}{\partial x_1} \geq 0, \quad \frac{\partial f}{\partial x_2} \geq 0, \quad \frac{\partial^2 f}{\partial x_1^2} \geq 0$$

Как известно из классической литературы по машинному обучению (например, из статьи Г. Цыбенко [7]), класс нейронных сетей с одним скрытым слоем и сигмодой, взятой в качестве функции активации, является универсальным аппроксиматором для непрерывных функций. Этот класс запишется в следующем виде:

$$N = \left\{ f(x) = w_0 + \sum_{i=1}^H w_i \cdot h \left( b_i + \sum_j v_{ij} x_j \right) \right\}$$

Эта запись не дает никаких гарантий на свойства  $f(x)$ , однако если заметить, что сигмоида является монотонно возрастающей функцией, то гарантировать положительность первых производных можно, добившись положительности весов. Самый простой способ сделать это – использовать вместо значений весов их экспоненты, что дает следующее выражение:

$$N_+ = \left\{ f(x) = e^{w_0} + \sum_{i=1}^H e^{w_i} \cdot h \left( b_i + \sum_j v_{ij} x_j \right) \right\} \quad (1)$$

Отдельно стоит отметить, что требовать положительность  $b_i$  не требуется, так как сигмоида от любого аргумента неотрицательна.

### 1.7.2 Архитектура, гарантирующая выпуклость цены

относительно  $\frac{S_0}{K}$

Так как сигмоида имеет положительную первую производную  $h'(x) = h(x)(1 - h(x))$ , то ее первообразная является выпуклой функцией. Обозначим ее как  $\zeta(\text{softplus})$ ,  $\zeta(x) = \ln(1 + e^x)$  и будем использовать там, где требуется выпуклая функция активации, а именно – для признака `money`, относительно которого цена и обладает свойством выпуклости. К остальным признакам будем применять сигмоиду, а затем перемножать получившиеся значения. Итоговая архитектура будет выглядеть так:

$$N_{++} = \left\{ f(x) = e^{w_0} + \sum_{i=1}^H e^{w_i} \left( \prod_{j=1}^c \zeta(b_{ij} + e^{\nu_{ij}} x_j) \right) \left( \prod_{j=c+1}^n h(b_{ij} + e^{\nu_{ij}} x_j) \right) \right\} \quad (2)$$

Далее две данные сети будут реализованы и обучены, результаты будут сравниваться с результатами метода Монте Карло.





## Глава 2

# Практическая часть

### 2.1 Реализация метода Монте Карло

Из теоретической части при постоянных  $r$  и  $\sigma$  имеем следующую запись:

$$S(T) = S(0) \exp \left[ \left( r - \frac{\sigma^2}{2} \right) T + \sigma \varepsilon \sqrt{T} \right]$$

Используем эту формулу для генерации траекторий цены базового актива. Пример сгенерированных траекторий можно видеть на графике ниже.

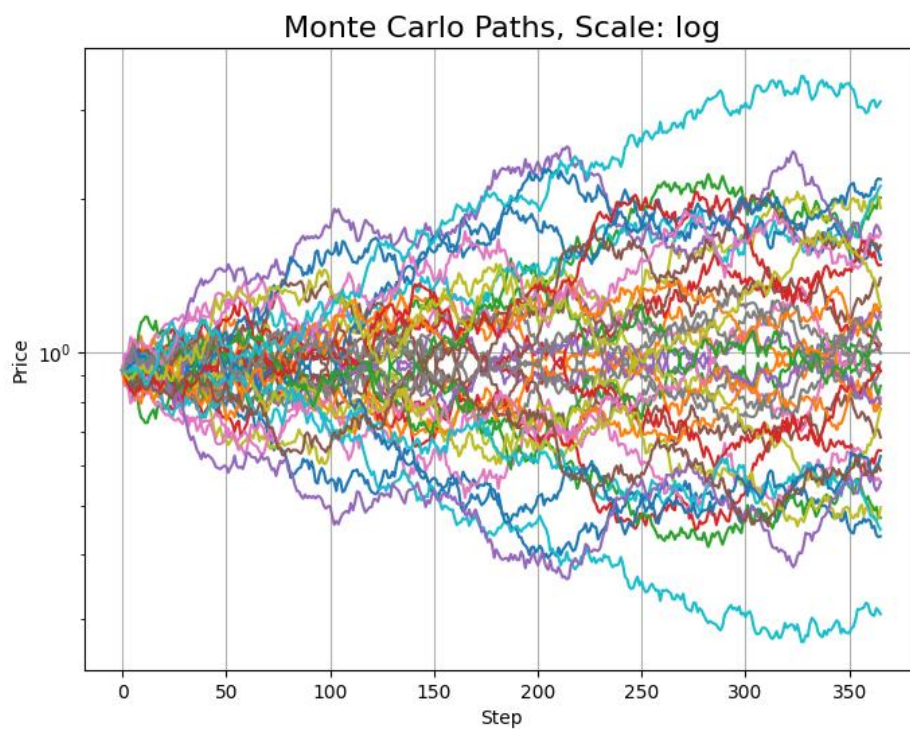


Рис. 3.

Стоит отметить, что при генерации траекторий на графике использовался метод antithetic variates. Это можно заметить, если посмотреть на наличие симметричных

относительно оси абсцисс траекторий. Для наглядности они изображены одним цветом.

Основное преимущество метода Монте Карло в том, что он может быть использован и когда цена опциона зависит только от финального значения  $S$ , и когда она зависит от траектории движения  $S$ . Легко вычислять цены инструментов, по которым происходит несколько выплат за период. К недостаткам метода можно отнести то, что он требует много вычислительных ресурсов, а также то, что он не может без модификаций работать с инструментами, по которым у покупателя есть право досрочного исполнения.

По алгоритму, описанному в теоретической части, рассчитаем цены опционов и доверительные интервалы для них. Пока не будем использовать метод antithetic variates. Укажем в таблице 1 первые 10 результатов.

Таблица 1. Цены и доверительные интервалы

Spot/Strike	TTM	Rate	Vol.	Avg type	Price/Strike	Left CI	Right CI
1.7006	0.7712	0.0503	0.3867	GEOM	0.6977	0.6271	0.7684
1.1763	0.7434	0.1320	0.1232	GEOM	0.2128	0.1980	0.2277
0.9228	1.4121	0.1239	0.4093	GEOM	0.0979	0.0619	0.1340
0.9480	1.4413	0.0351	0.3277	ARITHM	0.0832	0.0510	0.1154
1.2529	0.6457	0.1994	0.3754	GEOM	0.2973	0.2552	0.3395
1.6378	1.0749	0.1653	0.4035	ARITHM	0.6798	0.5964	0.7631
1.7836	1.3600	0.0542	0.1844	ARITHM	0.7947	0.7475	0.8418
0.7630	0.9878	0.1680	0.2958	ARITHM	0.0154	0.0054	0.0254
1.1791	0.7208	0.1686	0.4109	GEOM	0.2367	0.1931	0.2802
1.4761	0.7088	0.1335	0.2427	ARITHM	0.5017	0.4650	0.5385

Видим, что цены успешно рассчитаны, но доверительные интервалы достаточно широки. Для генерации были использованы следующие параметры:

1. Количество траекторий для каждой цены: 100
2. Количество отрезков времени: 365
3. Уровень значимости доверительных интервалов: 0.95

Не будем изменять параметры генерации, то теперь применим метод antithetic variates. Приведем аналогичную таблицу 2 с ценами и доверительными интервалами.

Видим, что исходные параметры (соотношение цены и страйка, ставка и др.) остались неизменными, рассчитанные цены изменились незначительно, а доверительные интервалы сузились в 3-4 раза. Можно сделать вывод, что метод Монте Карло лучше всего применять именно данной модификации.

Таблица 2. Цены и ДИ с Antithetic variates

Spot/Strike	TTM	Rate	Vol.	Avg type	Price/Strike	Left CI	Right CI
1.7006	0.7712	0.0503	0.3867	GEOM	0.6976	0.6857	0.7095
1.1763	0.7434	0.1320	0.1232	GEOM	0.2128	0.2121	0.2136
0.9228	1.4121	0.1239	0.4093	GEOM	0.0985	0.0763	0.1207
0.9480	1.4413	0.0351	0.3277	ARITHM	0.0831	0.0632	0.1031
1.2529	0.6457	0.1994	0.3754	GEOM	0.2976	0.2875	0.3077
1.6378	1.0749	0.1653	0.4035	ARITHM	0.6785	0.6571	0.6998
1.7836	1.3600	0.0542	0.1844	ARITHM	0.7946	0.7886	0.8006
0.7630	0.9878	0.1680	0.2958	ARITHM	0.0153	0.0083	0.0222
1.1791	0.7208	0.1686	0.4109	GEOM	0.2369	0.2219	0.2519
1.4761	0.7088	0.1335	0.2427	ARITHM	0.5017	0.4972	0.5061

## 2.2 Реализация и обучение нейронных сетей

Реализуем две архитектуры нейронных сетей. Сначала ту, которая гарантирует положительность цены и ее возрастание по параметру *moneyupness* (архитектура 1), а затем – ту, которая гарантирует выпуклость цены по нему (архитектура 2).

### 2.2.1 Архитектура, гарантирующая положительность и возрастание цены

Эта сеть реализует архитектуру 1 и состоит из двух слоев и функции активации – сигмоиды.

Первый слой – это модификация полносвязного слоя, которая отличается только тем, что в *forward pass* используется не сама матрица весов  $W$ , а  $e^W$ . Так как модификация используем встроенные в Pytorch (фреймворк для обучения моделей) векторные операции (в данном случае – функцию *torch.exp*), то градиенты вычисляются автоматически.

После функции активации (сигмоиды, как указано в статье [4]) следует следующая модификация линейного слоя. Теперь экспоненцируется не только матрица весов, то и вектор свободных членов, что как раз и гарантирует положительность цены, а также положительность первых производных.

Обучим сеть на датасете, полученном методом Монте Карло, зафиксировав следующие гиперпараметры.

1. Размер датасета: 100000 цен
2. Количество эпох: 300
3. Batch size: 100

В качестве функции потерь выберем *MSELoss*, а в качестве оптимизатора – Adam с параметрами  $lr = 3e^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\varepsilon = 1e^{-8}$ . Отложим 10% датасета для

теста, а остальное разделим в соотношении 80% и 20% (т.е. 72% и 18% от исходного датасета) для обучения и валидации.

Обучение прошло успешно, переобучения обнаружено не было, loss удалось снизить до  $10^{-4}$ . Приведем график 4 функции потерь MSE по ходу обучения.

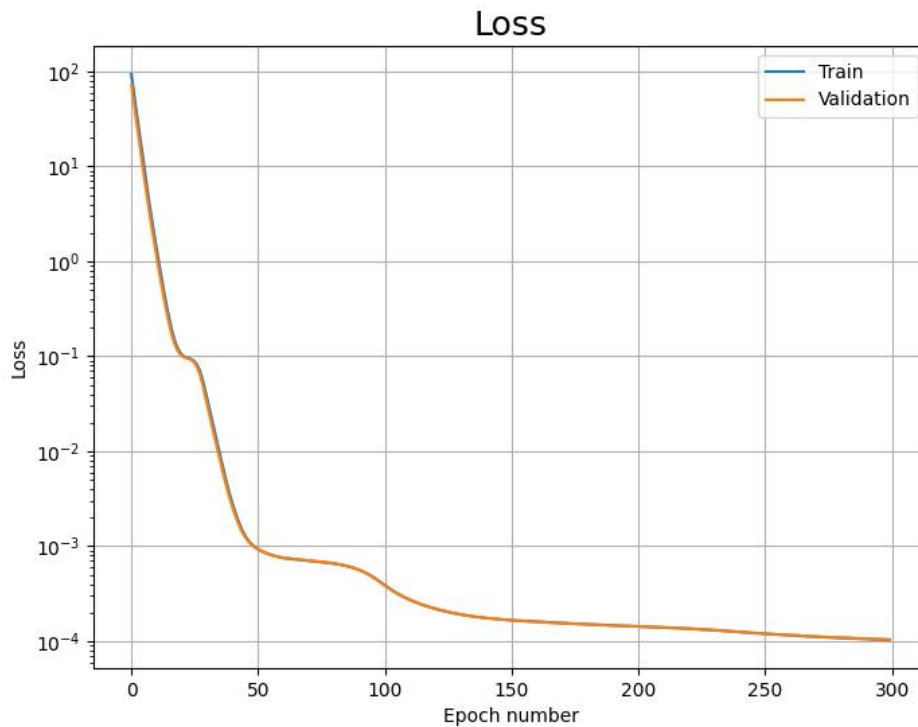


Рис. 4.

Цены, получившиеся в результате использования сети, представлены в таблице 3.

Таблица 3. Цены Монте Карло и первой сети

Spot/Strike	TTM	Rate	Vol.	Avg type	MC price	Net price	Is in CI
1.7006	0.7712	0.0503	0.3867	GEOM	0.6976	0.6980	+
1.1763	0.7434	0.1320	0.1232	GEOM	0.2128	0.1894	-
0.9228	1.4121	0.1239	0.4093	GEOM	0.0985	0.0918	+
0.9480	1.4413	0.0351	0.3277	ARITHM	0.0831	0.0814	+
1.2529	0.6457	0.1994	0.3754	GEOM	0.2976	0.3120	-
1.6378	1.0749	0.1653	0.4035	ARITHM	0.6785	0.6727	+
1.7836	1.3600	0.0542	0.1844	ARITHM	0.7946	0.8005	+
0.7630	0.9878	0.1680	0.2958	ARITHM	0.0153	0.0214	+
1.1791	0.7208	0.1686	0.4109	GEOM	0.2369	0.2517	+
1.4761	0.7088	0.1335	0.2427	ARITHM	0.5017	0.5101	-

Цены, полученные в результате использования сети, достаточно близки к ценам Монте Карло. Они попадают в 95% доверительный интервал с использованием antithetic variates в 7 случаях из 10.

Расчет 10000 цен в режиме inference занял 543 мс, в то время как аналогичная задача, решаемая с помощью метода Монте Карло, заняла 20.558 с. Без учета времени на обучение сети получаем экономию в 38 раз.

Теперь убедимся в том, что данная архитектура действительно гарантирует неотрицательность и возрастание цены. Для этого сгенерируем следующий датасет. Зафиксируем все параметры, кроме *moneyupness*, который будем равномерно варьировать в диапазоне от 0.5 до 1.5. Для определенности установим  $ttm = 1$ ,  $rate = 0.1$ ,  $volatility = 0.2$ , тип усреднения – арифметический. Сначала визуализируем результаты Монте Карло, чтобы убедиться в том, что исходный датасет корректен.



Рис. 5.

Видим, что оба предположения (а также и предположение о выпуклости, которое потребуется позднее) выполняются. Тогда обучим на этом датасете (размер – 10000 цен) сеть так же, как обучали в общем случае, когда параметры не были фиксированы. В силу меньшего размера датасета будем обучать сеть в течение 1000 эпох вместо 300. Получим следующий график 6. Теперь мы убедились в том, что предположения о положительности и возрастания выполняются. Однако при больших значениях параметра *moneyupness* функция становится вогнутой, что противоречит свойству цен опционов. Именно этот недостаток позволяет исправить следующая архитектура.

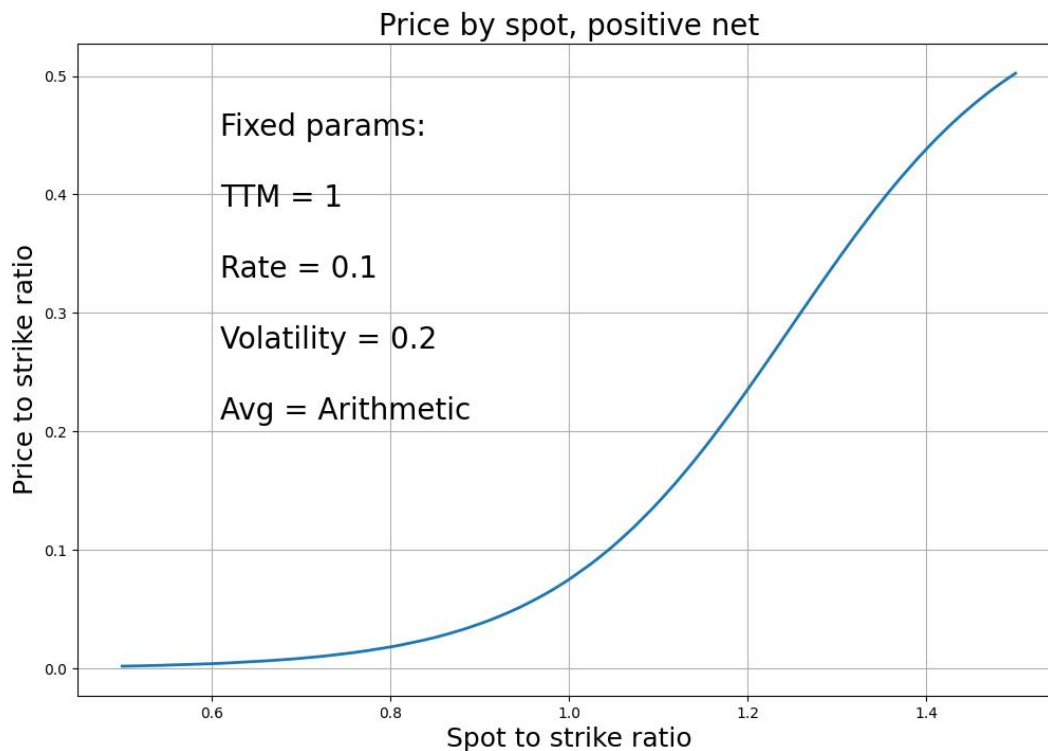


Рис. 6.

### 2.2.2 Архитектура, гарантирующая выпуклость цены относительно $\frac{S_0}{K}$

Эта сеть состоит из двух слоев и функции активации. Первый слой применяет линейное преобразование к входным данным, причем, как и в случае предыдущей сети, матрица весов входит в формулу в виде  $e^W$ . Особенностью данного слоя является то, что не происходит суммирования по признакам после умножения входного вектора на матрицу весов. Вместо этого к каждому признаку применяется своя функция активации в зависимости от того, является ли результат (цена) выпуклой по данному признаку или нет, после чего полученные значения перемножаются. К признакам, по которым есть выпуклость, применяется выпуклая функция активации  $\text{softplus} = \ln(1 + e^x)$ , а к остальным – сигмоида  $h(x) = \frac{1}{1+e^{-x}}$ .

Третий слой – это модификация линейного слоя, такая же, как и в последнем слое предыдущей сети. Используются экспоненты как от матрицы весов, так и от вектора свободных членов.

Обучим сеть на том же датасете, зафиксировав следующие гиперпараметры.

1. Размер датасета: 100000 цен
2. Количество эпох: 300
3. Batch size: 100

В качестве функции потерь выберем  $\text{MSELoss}$ , а в качестве оптимизатора – Adam с параметрами  $lr = 3e^{-4}$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\varepsilon = 1e^{-8}$ . Отложим 10% датасета для теста, а остальное разделим в соотношении 80% и 20% (т.е. 72% и 18% от исходного датасета) для обучения и валидации.

Обучение прошло успешно, переобучения обнаружено не было, loss удалось снизить ниже  $10^{-3}$ . Приведем график 7 функции потерь MSE по ходу обучения.

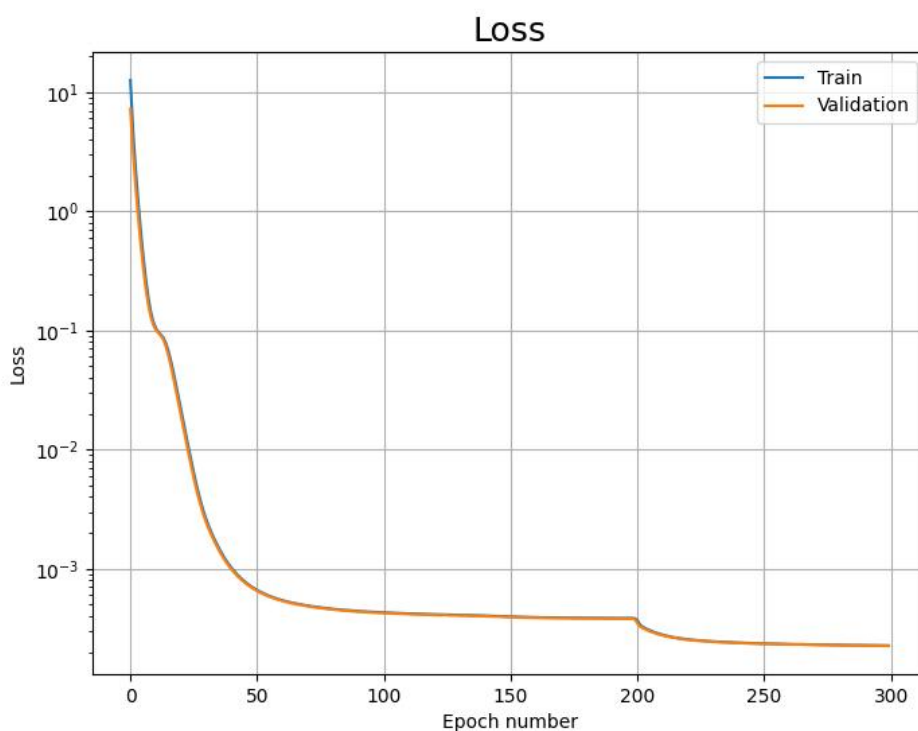


Рис. 7.

Цены, получившиеся в результате использования выпуклой сети, представлены в таблице 4

Таблица 4. Цены Монте Карло и выпуклой сети

Spot/Strike	TTM	Rate	Vol.	Avg type	MC price	Net price	Is in CI
1.7006	0.7712	0.0503	0.3867	GEOM	0.6976	0.6994	+
1.1763	0.7434	0.1320	0.1232	GEOM	0.2128	0.2178	-
0.9228	1.4121	0.1239	0.4093	GEOM	0.0985	0.0697	-
0.9480	1.4413	0.0351	0.3277	ARITHM	0.0831	0.0769	+
1.2529	0.6457	0.1994	0.3754	GEOM	0.2976	0.2920	+
1.6378	1.0749	0.1653	0.4035	ARITHM	0.6785	0.6838	+
1.7836	1.3600	0.0542	0.1844	ARITHM	0.7946	0.7902	+
0.7630	0.9878	0.1680	0.2958	ARITHM	0.0153	0.0220	+
1.1791	0.7208	0.1686	0.4109	GEOM	0.2369	0.2278	+
1.4761	0.7088	0.1335	0.2427	ARITHM	0.5017	0.5016	+

Цены, полученные в результате использования выпуклой сети, ближе к ценам

Монте Карло, чем цены сети в предыдущем случае. Они попадают в 95% доверительный интервал с использованием antithetic variates в 8 случаях из 10 (против 7 случаев в предыдущей сети).

Расчет 10000 цен в режиме inference занял 4.125 с, в то время как аналогичная задача, решаемая с помощью метода Монте Карло, заняла 20.558 с. Без учета времени обучения сети получаем экономию времени в 5 раз. По сравнению с сетью, не гарантирующей выпуклость, имеем замедление, связанное с более сложными вычислениями.

Теперь убедимся в том, что данная архитектура действительно гарантирует неотрицательность, возрастание и выпуклость цены относительно параметра moneyness. Обучим данную сеть на датасете из аналогичной задачи для предыдущей архитектуры, где были фиксированы все параметры, кроме moneyness. Получим следующий график 8.

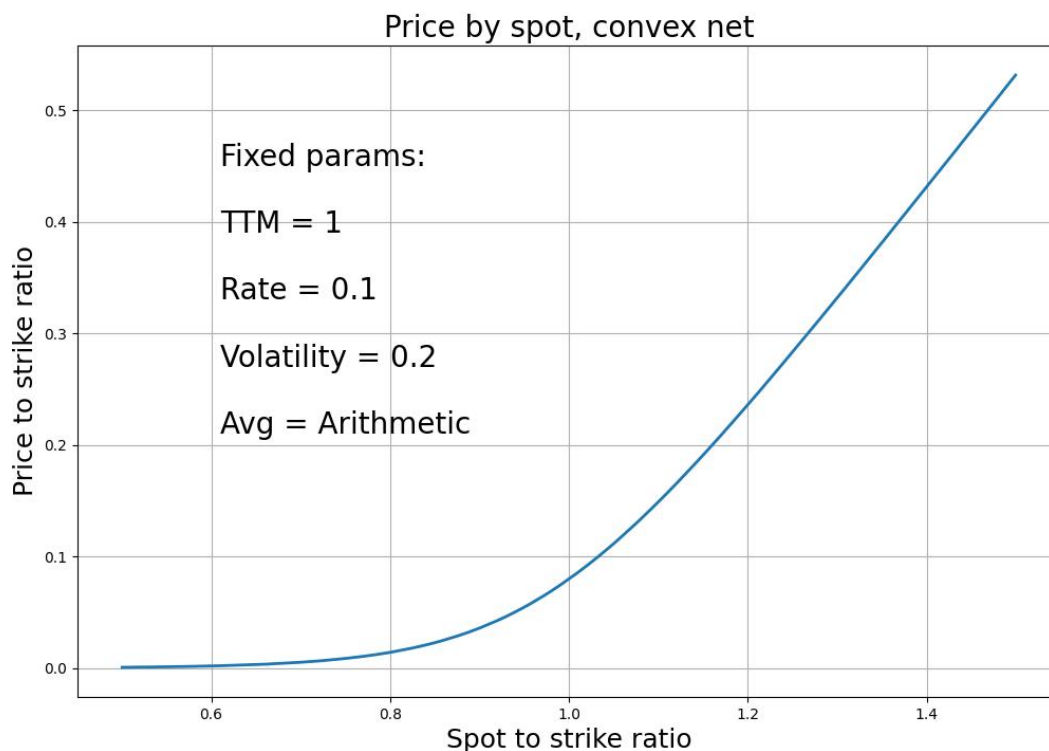


Рис. 8.

Теперь мы убедились в том, что предположения о положительности, возрастания и выпуклости цены выполняются при использовании данной архитектуры.

## 2.3 Сравнение результатов

На графиках, на которых показано убывание функции потерь по время обучения, видно, что изначальное ее значение на порядок ниже для выпуклой сети, чем



для невыпуклой ( $10^1$  против  $10^2$ ). Приведем более подробные данные, добавив также разделение на тип усреднения. Функция цены не является непрерывной по этому параметру, поэтому, возможно, разделение арифметических и геометрических опционов позволит достичь лучшего качества.

Обучим 6 сетей на датасете размером 500 цен, количество эпох – 300. Три выпуклых и три невыпуклых, в каждом наборе – только арифметические опционы, только геометрические или и те, и другие, тогда используем тип усреднения в качестве параметра. На графике 9 ниже представлена зависимость значения функции потерь от эпохи для каждой из сетей.

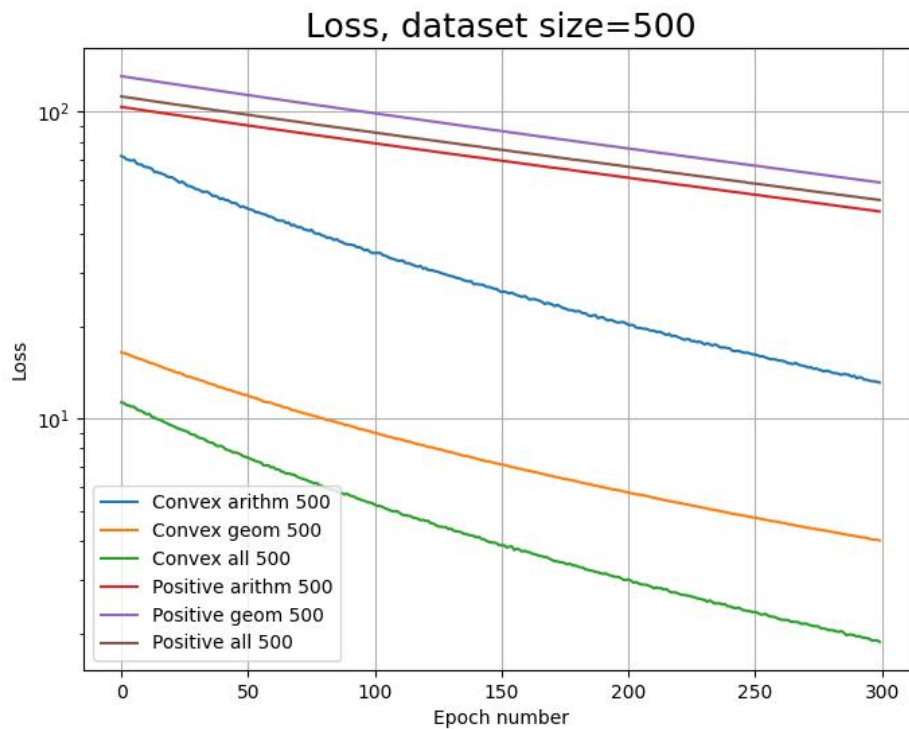


Рис. 9.

Можно заметить, что основным фактором, определяющим величину функции потерь, является тип сети. Лучше всего обучается выпуклая сеть, работающая со всеми опционами без разделения на арифметические и геометрические. Хуже всего – невыпуклая сеть с исключительно геометрическими опционами. В связи с этим, можно заключить, что разделение опционов не помогает решать задачу.

## 2.4 Выводы

В ходе исследования были реализованы три метода расчета цен азиатских опционов: с использованием метода Монте Карло и с использованием двух архитектур нейронных сетей, каждая из которых давала свои гарантии относительно цены.

Для метода Монте Карло хорошо себя зарекомендовал метод снижения дисперсии – использование antithetic variates. С его помощью ширина доверительного интервала была снижена в 3-4 раза без генерации дополнительных траекторий.

Применение машинного обучения позволило снизить временные затраты на расчет цен в 5 раз при использовании архитектуры, гарантирующей выпуклость цены относительно параметра `moneyiness`, и в 38 раз при использовании архитектуры, гарантирующей только положительность и возрастание цены.

Проведенный анализ скорости обучения позволяет заключить, что на небольшом размере датасета и ограниченных ресурсах очень желательно использовать выпуклую сеть, так как ее функция потерь в начале обучения на порядок меньше, чем функция потерь невыпуклой сети. При большом количестве данных следует выбирать между гарантиями и количеством доступных ресурсов. Так, использование выпуклой сети существенно замедляет обучение (1377 секунд против 90.8 секунд для невыпуклой сети для датасета из 100.000 цен и при обучении в течение 300 эпох) из-за более сложной логики, но не дает значительного выигрыша в точности. Также выпуклая сеть работает медленнее невыпуклой в режиме `inference`. Тем не менее, эта архитектура позволяет отсеять экономически некорректные результаты, в которых цена не является выпуклой функцией.

# Литература

- [1] Zhou Fang. “Application of machine learning: an analysis of asian options using neural network”. *Oklahoma State University* [https://shareok.org/bitstream/handle/11244/300104/Fang\\_okstate\\_0664M\\_15329.pdf?sequence=1&isAllowed=y](https://shareok.org/bitstream/handle/11244/300104/Fang_okstate_0664M_15329.pdf?sequence=1&isAllowed=y) (Дата обращения: 17.03.2022)
- [2] Yi Dong. “Accelerating Python for Exotic Option Pricing”. <https://developer.nvidia.com/blog/accelerating-python-for-exotic-option-pricing/> (Дата обращения: 17.03.2022)
- [3] Boris Hanin. “Universal function approximation by deep neural nets with bounded width and ReLU activations” <https://arxiv.org/abs/1708.02691v3> (Дата обращения: 17.03.2022)
- [4] Charles Dugas. “Incorporating Functional Knowledge in Neural Networks” *Journal of Machine Learning Research* <https://jmlr.csail.mit.edu/papers/volume10/dugas09a/dugas09a.pdf> (Дата обращения: 21.02.2022)
- [5] John C. Hull. “Options, futures and other derivatives” *University of Toronto*, 9th edition, 2015
- [6] Chang Ma, Mr. Fabian Valencia. “Welfare Gains from Market Insurance: The Case of Mexican Oil Price Risk” *International Monetary Fund* <https://www.elibrary.imf.org/view/journals/001/2018/035/001.2018.issue-035-en.xml> (Дата обращения: 07.04.2022)
- [7] Cybenko, G. “Approximation by superpositions of a sigmoidal function. Math”. *Control Signal Systems* 2, 303–314 (1989) <https://link.springer.com/article/10.1007/BF02551274> (Дата обращения: 08.04.2022).
- [8] Kemna, A.G.Z., Vorst, A.C.F. “A pricing method for options based on average asset values”. *Journal of Banking and Finance* (1990), 14, 113-129.
- [9] Levy, E. “Pricing European Average Rate Currency Options”, *Journal of International Money and Finance* (1992), 11, 474-491.

- [10] Wiklund, E. “Asian Option Pricing and Volatility”. <https://www.math.kth.se/matstat/seminarier/reports/M-exjobb12/120412a.pdf> (Дата обращения: 09.04.2022)
  
- [11] Contract specifications. London Metal Exchange. <https://www.lme.com/en/Metals/Non-ferrous/LME-Aluminium/Contract-specifications> (Дата обращения: 09.04.2022)