



UNIVERSIDAD NACIONAL DE CÓRDOBA
FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y COMPUTACIÓN.

Métodos computacionales para el cálculo de la volatilidad implícita del modelo de Black Scholes

Tesis realizada por Diego Juan Nasareno Lupi para la Licenciatura en
Ciencias de la Computación en la Universidad Nacional de Córdoba

Dirigida por:
Dra. Noemí Patricia Kisbye
Lic. Pedro Ángel Pury



Esta obra está bajo una [Licencia Creative Commons Atribución-NoComercial 4.0 Inter-
nacional](https://creativecommons.org/licenses/by-nc/4.0/).

Agradecimientos

- A mi familia, por el apoyo durante toda la carrera, en especial mi hermano Emanuel.
- A los todos los docentes que he tenido durante la carrera, en especial a Pedro y Patricia por su paciencia y dedicación.
- A mis amigos y compañeros.

Resumen

Las finanzas cuantitativas constituyen, desde hace varias décadas, un área particular de estudio dentro de la matemática. Esta nueva disciplina surge de la necesidad de encontrar modelos cuantitativos que permitan describir el comportamiento aleatorio de activos financieros y, en particular, valorar los productos llamados derivados financieros.

Si la hipótesis sobre la dinámica de los activos es que estos siguen un proceso estocástico lognormal, con tendencia y volatilidad constante, entonces la valoración de una opción call sobre el activo está dada por la fórmula de Black-Scholes. Ahora bien, dado que la volatilidad no es observable en el mercado, se define la volatilidad implícita del activo como aquella que iguala la prima del mercado con el valor dado por la fórmula.

La obtención de este parámetro de volatilidad implícita permite luego valorar otros derivados financieros como así también comprender movimientos propios del mercado.

Por otra parte, la determinación de la volatilidad implícita requiere de la aplicación de métodos numéricos, puesto que se trata de resolver una ecuación no lineal sin una solución cerrada. En los últimos años, a su vez, han aparecido propuestas de uso de métodos de aprendizaje automático para modelar de manera empírica la función que provee la volatilidad implícita.

Este trabajo incluye la exploración bibliográfica referida al concepto de volatilidad implícita y sus implicancias, y de métodos computacionales factibles de ser implementados para su cálculo. Además se realizará la implementación efectiva en computadora de algunas soluciones y se hará un análisis comparativo de la eficiencia de los distintos métodos estudiados.

Índice general

Agradecimientos	3
Resumen	5
Índice general	7
1. Introducción	9
1.1. Motivación	9
1.2. Objetivos del trabajo	9
1.3. Estructura de la Tesis	10
2. Nociones Preliminares	11
2.1. Conceptos financieros elementales	11
2.1.1. Los tipos de Interés	11
2.1.2. Tasa libre de riesgo	12
2.1.3. Productos básicos	12
2.1.4. Derivados	12
2.1.5. Mercados financieros	12
2.1.6. Opciones	13
2.1.7. Payoff y Ganancia	14
2.1.8. Estrategias con opciones	15
2.1.9. Modelo de Black-Scholes	18
2.1.10. Fórmula de Valoración de una call europea con Black-Scholes	19
2.1.11. Volatilidad	19
2.1.12. Superficie de volatilidad	21
2.2. Método de Bisección	21
2.3. Método de Brent	22
2.4. Redes neuronales	23
2.4.1. Perceptrón	23
2.4.2. Redes Feed-Forward	25
2.4.3. Aprendizaje por Descenso por el Gradiente	26

2.4.4. Funciones de Activación	27
2.4.5. Tasa de Aprendizaje	28
2.5. k -fold cross validation	29
3. Implicancias de la volatilidad implícita	31
3.1. Superficie de Volatilidad para valorar opciones	31
3.2. Estrategias dependiendo de la volatilidad implícita	32
3.2.1. Backspread	33
3.2.2. Long Straddle y Long Strangle	34
3.2.3. Long Butterfly Spread	35
4. Implementación numérica del método de Bisección y de Brent	39
4.1. Introducción	39
4.2. Implementación de los métodos numéricos	40
4.2.1. Intervalo inicial para el método de bisección y de Brent	40
4.2.2. Aplicación de los Métodos Numéricos	40
5. Implementación con Red Neuronal Feed-Forward	41
5.1. Cálculo de prima de opción call usando S/K	41
5.2. Generación de muestra	41
5.3. k -fold cross validation	43
5.4. Optimización	48
6. Resultados	49
6.1. Red inicial	49
6.2. Optimización	50
6.2.1. Cambio en función de decrecimiento de la Tasa de Aprendizaje	51
6.3. Métodos Numéricos	54
6.3.1. Problemas Numéricos	54
6.3.2. Precisión Métodos Numéricos	55
6.4. Tiempo de Ejecución y Robustez	58
7. Conclusiones	61
Bibliografía	63

Capítulo 1

Introducción

Para el desarrollo del presente trabajo es necesario estudiar y comprender varias herramientas y técnicas que usaremos a lo largo del proyecto. A continuación presentamos un resumen de los temas que abordaremos.

1.1. Motivación

La volatilidad implícita en estos días ha adquirido una gran importancia. Se puede utilizar para el cálculo de superficies de volatilidad. Las superficies de volatilidad son utilizadas para valorar opciones mediante modelos matemáticos.

Otra utilidad es como parámetro de riesgo para inversores, dependiendo de la volatilidad implícita se pondera algunas estrategias sobre otras para disminuir el riesgo y aumentar su posibilidad de rentabilidad.

Por eso es indispensable poder calcular la volatilidad implícita, generalmente se utilizan métodos iterativos para estimarla, pero estos métodos suelen ser lentos. Por lo tanto se propone el cálculo de la volatilidad implícita mediante redes neuronales para disminuir considerablemente el tiempo de cálculo siguiendo como guía el artículo [1], además cada vez es mayor el uso de redes neuronales en el campo de las finanzas, por ejemplo [2], [3].

1.2. Objetivos del trabajo

El objetivo del trabajo es determinar la volatilidad implícita sobre opciones call europeas mediante métodos numéricos, o el uso de métodos de aprendizaje automático para modelar de manera empírica la función que provee la volatilidad implícita; ya que se trata de resolver una ecuación no lineal sin una solución cerrada.

1.3. Estructura de la Tesis

La tesis está estructurada de la siguiente manera. En el capítulo 2 se presentan los conceptos matemáticos, financieros y técnicos utilizados a lo largo del trabajo. En el capítulo 3 se muestra las implicancias de la volatilidad implícita, y la importancia de su cálculo. Los capítulos 4 y 5 refieren a la implementación de métodos para el calculo de la volatilidad implícita, tanto métodos numéricos (Bisección y Brent), como redes neuronales. En el capítulo 6 se presentan los resultados obtenidos comparando los métodos numéricos con la red neuronal. Y por último en el capítulo 7 se presenta una breve conclusión y posibles mejoras.

Capítulo 2

Nociones Preliminares

2.1. Conceptos financieros elementales

2.1.1. Los tipos de Interés

Si depositamos dinero en una cuenta bancaria, al cabo de un cierto tiempo este capital se incrementa en un determinado monto, llamado **interés**.

1. **Interés Simple:** La fórmula correspondiente es:

$$V = V_0(1 + rt)$$

donde V representa el valor de un depósito de valor inicial V_0 transcurrido un tiempo t y r es la tasa de interés correspondiente a la unidad de tiempo utilizada, usualmente el año.

2. **Interés Compuesto:**

$$V = V_0 \left(1 + \frac{r}{n}\right)^{nt}$$

donde que V , V_0 y t tienen las mismas condiciones que para el interés simple, pero una tasa r de interés nominal anual compuesta n veces al año.

3. **Interés Continuo:**

El interés continuo puede verse como el interés compuesto para $n \rightarrow \infty$, es decir:

$$V = \lim_{n \rightarrow \infty} V_0 \left(1 + \frac{r}{n}\right)^{nt} = V_0 e^{rt}$$

2.1.2. Tasa libre de riesgo

A los efectos de valoración de derivados se asume la existencia de una tasa llamada **tasa libre de riesgo**. Se trata de una tasa de referencia que no tiene riesgo crediticio, es decir, que un inversor sabe que invirtiendo a esa tasa podrá recuperar el capital. Por ejemplo, los bonos del tesoro (Bonar 2025), o alguna tasa a la cual el propio estado ofrece para la devolución del préstamo (LEBAC, LETES) [4].

2.1.3. Productos básicos

Denominamos productos básicos a aquellos instrumentos financieros cuyo valor no depende de otro activo. Entre ellos están las acciones, los índices, las monedas, los commodities y los bonos.

2.1.4. Derivados

Un derivado puede ser definido como un instrumento financiero cuyo valor depende o deriva de los valores de otros activos subyacentes. Estos activos subyacentes podrían ser activos básicos u otros derivados.

En términos generales, un derivado consiste en un contrato entre dos partes para comprar o vender el subyacente. Las características y condiciones de estos contratos dan lugar a diferentes tipos de derivados, tales como contratos forward, futuros y opciones.

2.1.5. Mercados financieros

Los instrumentos financieros se comercializan en el mercado financiero. En la práctica existe un mercado formal u organizado y un mercado extrabursátil, denominado también *over the counter market* (mercados OTC). En el caso del mercado formal, las negociaciones son multilaterales, es decir, existen múltiples agentes que actúan como compradores o vendedores de productos financieros. Los instrumentos que se comercializan están estandarizados, y existe una regulación de este tipo de mercados para proteger a los inversores de posibles incumplimientos de las contrapartes. En el caso del mercado extrabursátil, las negociaciones son bilaterales, es decir, entre dos partes. En estos casos los contratos suelen acordarse entre las partes en cuanto a la cantidad y características del subyacente. No existe una regulación formal sino que se basa en un principio de confianza entre partes.

El tamaño del segmento de mercados OTC es varias veces mayor que el de los mercados regulados, especialmente en los contratos de tipos de interés y de divisas. Según un estudio de ICAP[5], estima que cada día se producen 2 millones de transacciones en los mercados OTC por un nominal de 5 billones de dólares.

2.1.6. Opciones

Las opciones son contratos que dan derecho a una de sus partes a comprar (o vender) el subyacente, a un precio determinando en un tiempo futuro. Las opciones que dan derecho a compra se denominan **calls** y las que dan derecho a venta se denominan **puts**. Estos contratos tienen un valor inicial denominado prima, que es el precio al cual compra el contrato quien adquiere el derecho a compra o venta.

Quien compra una opción está en posición long sobre el contrato, y quien la vende está en posición short.

El agente que esté en posición long tiene derecho a ejercerla o no. En cambio el que este en posición short tiene una obligación sobre lo que haga su contraparte.

Las opciones que se negocian en mercados formales se denominan *opciones vanilla o estándar*.

Dentro de las opciones vanilla existen dos tipos:

- **Opciones Europeas:** son aquellas cuyo ejercicio ocurre sólo en la fecha de madurez.
- **Opciones americanas:** son aquellas que pueden ser ejercidas en cualquier momento previo a la madurez.

En el mercado OTC se negocia una variedad mucho mayor de opciones y se denominan en general *opciones exóticas* [4]. Dentro de las *opciones exóticas* se agrupan a todas aquellas opciones que tienen una mayor complejidad en comparación con las opciones vanilla. Algunas opciones exóticas son:

- **Opciones lookback:** son aquellas cuyo payoff depende del valor máximo, o del valor mínimo, que haya alcanzado el subyacente desde el inicio del contrato hasta su madurez.
- **Opciones asiáticas:** son aquellas cuyo payoff depende del promedio de valores que ha tomado el subyacente durante la vigencia del contrato, o de una parte de ese tiempo.
- **Opciones barrera:** son aquellas cuyo payoff depende de que el subyacente haya cruzado una determinada barrera a lo largo de la vigencia del contrato.

Otras opciones exóticas son las binarias, bermudas, choice, shout, basket, exchange, y muchas otras.

En cada contrato se fija entonces un precio de ejercicio o **strike**, que es el precio pactado al cual se comprará o venderá el subyacente, en la fecha de expiración o también llamada **madurez** del contrato. Así, un inversor que negocia una opción adquiere una de las siguientes posiciones en el contrato:

- **posición long:** quien compra la opción, y por lo tanto tiene derecho a ejercerla.
- **posición short:** quien vende la opción o suscriptor, y por lo tanto contrae una obligación.

2.1.7. Payoff y Ganancia

El payoff de una opción europea es el valor del contrato en su madurez en función del valor del subyacente. El costo inicial o prima de la opción no se incluye en el cálculo. Si K es el precio strike y $S(T)$ es el precio final del subyacente, entonces una opción call se ejerce sólo si $S(T) > K$ ya que en caso contrario el inversor en posición long preferirá comprar el subyacente en el mercado. En el caso de una put, el inversor en posición long ejercerá la opción sólo si $K > S(T)$. Así, el valor del contrato en su madurez está dado por la ganancia o pérdida neta del inversor en caso de que se ejerza o no la opción. El Cuadro 2.1 resume los payoff de las opciones call y put europeas. Observar las Figuras 2.1 2.2.

Los diagramas de payoff para cada una de las posiciones en una opción europea con strike K son los gráficos del Payoff en función del precio final del activo subyacente. El beneficio o ganancia real del inversor incluye además el costo de la prima (Figuras 2.3 2.4).

Payoff \pm prima

donde \pm dependerá de la posición long $(-)$ o la posición short $(+)$ [4].

Cuadro 2.1: Payoffs de call y put europeas

Opción	Payoff
Long en una Call	$\max(S(T) - K, 0)$
Short en una Call	$-\max(S(T) - K, 0) = \min(K - S(T), 0)$
Long en una Put	$\max(K - S(T), 0)$
Short en una Put	$-\max(K - S(T), 0) = \min(S(T) - K, 0)$

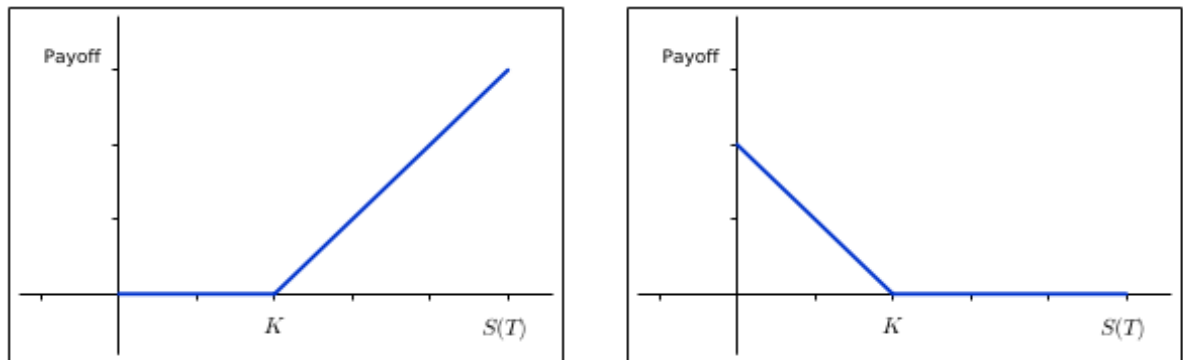


Figura 2.1: Payoff de posiciones long en una call y en una put con strike K y madurez T .

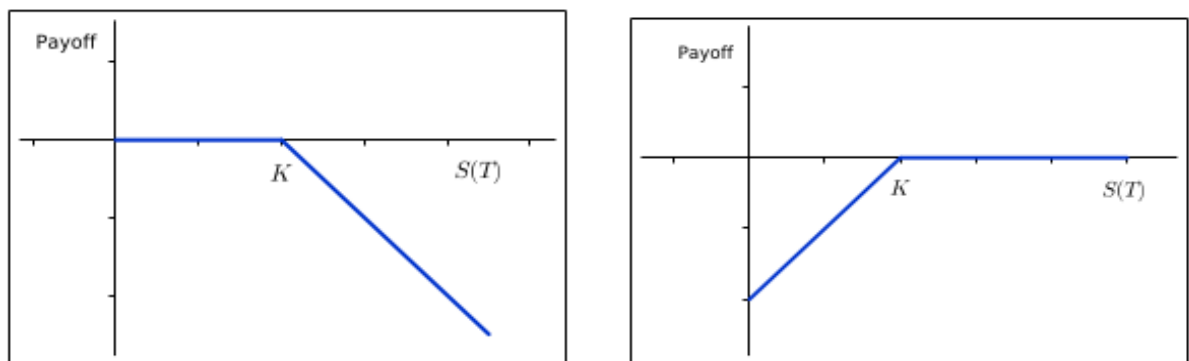


Figura 2.2: Payoff de posiciones short en una call y en una put con strike K y madurez T .

2.1.8. Estrategias con opciones

Las estrategias son mecanismos que utilizan los inversionistas para obtener ganancia, o cubrirse de posibles casos adversos. A la combinación de diferentes opciones y activos se las llama **portfolio**.

Un **portfolio** o cartera, es un conjunto de activos, derivados e inversiones bancarias que puede tener un inversor. El valor de este portfolio es la suma de los valores de sus componentes, con un signo positivo o negativo según estén a favor o no del inversor. Las posiciones long en un activo o derivado financiero y una cuenta bancaria son valores positivos. Las posiciones short y las deudas tienen valor negativo.

Veremos algunos portfolios específicos contruidos con opciones.

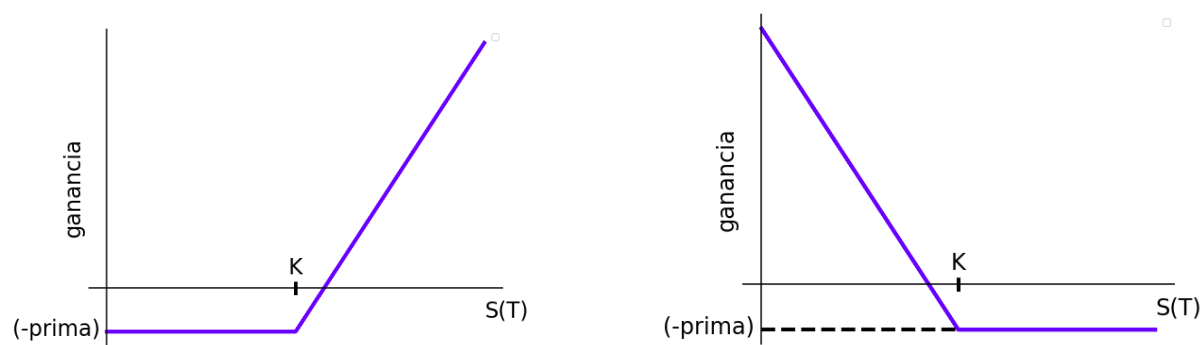


Figura 2.3: Ganancia de posiciones long en una call y en una put con strike K y madurez T .

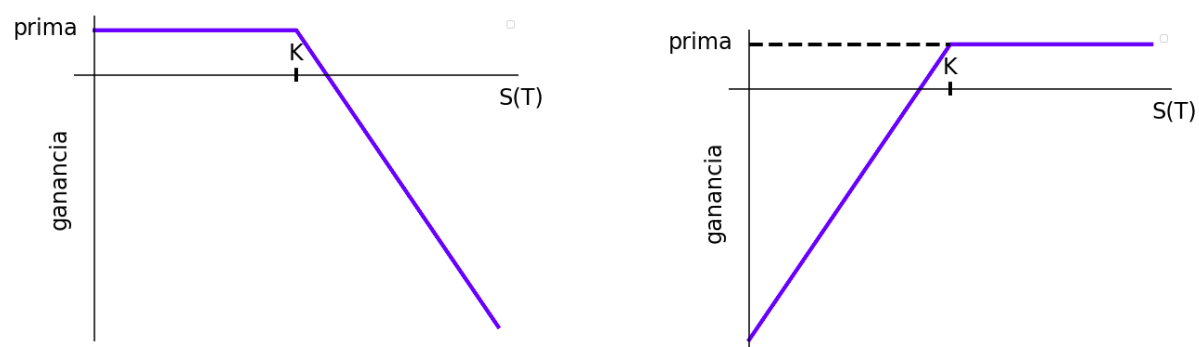


Figura 2.4: Ganancia de posiciones short en una call y en una put con strike K y madurez T .

Spreads

Los *spreads* son estrategias que utilizan opciones del mismo tipo. Esto es: todas call o todas put. Un inversor que cree que el mercado está en alza utiliza un *bull spread*, mientras que si cree que está en baja utiliza un *bear spread*. Un *bull spread* con calls se construye con una posición long en una call, con prima c_1 y strike K_1 , y 1 posición short en una call, con prima c_2 strike K_2 , con $K_1 < K_2$ y ambos con la misma fecha de expiración T . Luego el payoff está dado por:

$$\text{Payoff}(\text{bull spread con calls}) = \max\{S(T) - K_1, 0\} - \max\{S(T) - K_2, 0\}$$

Notemos que el payoff es no negativo, y su valor es positivo si $S(T)$ es mayor que K_1 . La ganancia se observa en la Figura 2.5.

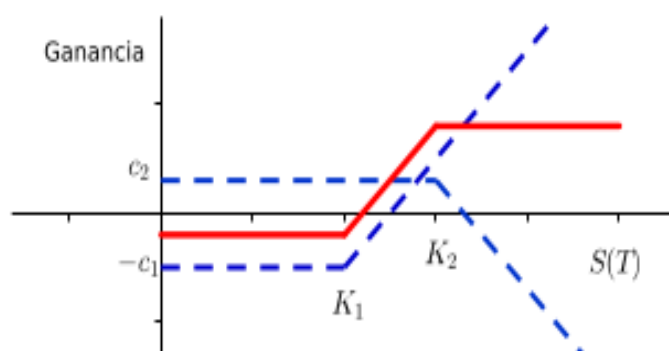


Figura 2.5: Diagrama de ganancia del bull spread con calls

Sin embargo, un *bull spread* con calls tiene un costo inicial $c_2 - c_1$. Por ello, si el valor de la acción baja habrá una pérdida, aunque acotada.

Combinaciones

Las combinaciones, son estrategias que consisten en posiciones en distintos tipos de opción, tanto call como puts. Por ejemplo, un *straddle* se construye con una posición long en una call con prima c y posición long en una put con prima p , ambos con la misma fecha de expiración T y strike K .

El payoff de una *straddle* está dado por:

$$\text{Payoff}(\text{bull spread con calls}) = \max\{S(T) - K, 0\} + \max\{K - S(T), 0\} = |S(T) - K|.$$

La Figura 2.6 muestra la ganancia del straddle. Notemos que el costo inicial está dado por $c + p$.

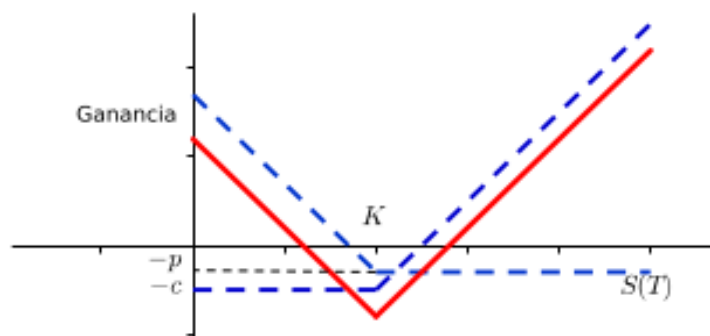


Figura 2.6: Diagrama de ganancia para un straddle

2.1.9. Modelo de Black-Scholes

Esta fórmula comienza con el diseño de un modelo, denominado Modelo de Black-Scholes, en el cual describen el comportamiento de los precios de las acciones, bajo ciertas hipótesis de mercado. A partir de esto, logran derivar una ecuación diferencial, la ecuación de Black-Scholes que es satisfecha por las opciones call y put europeas, y cuya resolución permite dar el valor exacto de la prima de la opción en un escenario de mercado sin arbitraje. La solución de esta ecuación diferencial es conocida como la Fórmula de Black-Scholes.

Los supuestos hechos por Black y Scholes cuando derivaron su fórmula de valoración de opciones fue el siguiente:

- no arbitraje: No es posible invertir en un portfolio con costo cero y que exista una probabilidad positiva de ganancia futura y una probabilidad nula de pérdida.
- completo (Todo derivado puede ser replicable con un portfolio compuesto por el subyacente y la cuenta bancaria.)
- precio de un activo que sigue un Movimiento Geométrico Browniano [4].
- Tasa de interés libre de riesgo constante.
- Posibilidad de invertir en un número ilimitado de opciones y acciones, y de manera continua.

2.1.10. Fórmula de Valoración de una call europea con Black-Scholes

La hipótesis del modelo de Black-Scholes supone que el valor del activo $S(t)$ se comporta de acuerdo a un movimiento geométrico browniano, con tendencia μ y volatilidad σ . Sin embargo se demuestra que, bajo una hipótesis de no arbitraje es posible realizar un cambio de medida de probabilidad bajo la cual $S(t)$ sigue un movimiento geométrico browniano con tendencia r y volatilidad σ , siendo r la tasa libre de riesgo con capitalización continua. Esta medida de probabilidad también se **llama medida de probabilidad neutral al riesgo**

Consideremos que c es la prima de una opción call europea, con strike K y madurez T , sobre un activo cuyo precio sigue un movimiento geométrico browniano con tendencia r y volatilidad σ bajo las probabilidades de riesgo neutral. Sea r la tasa libre de riesgo. Entonces, bajo una hipótesis de no arbitraje se cumple que:

$$c = S(0)\Phi(d_1) - Ke^{-rT}\Phi(d_2) \quad (2.1)$$

donde

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}t^2} dt \quad (2.2)$$

$$d_1 = \frac{\ln\left(\frac{S(0)}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \quad (2.3)$$

$$d_2 = d_1 - \sigma\sqrt{T} \quad (2.4)$$

con $S(0)$ el precio inicial del activo, r la tasa libre de riesgo, K el strike, T la madurez de la opción y σ la volatilidad de la opción [6].

Para simplificar la notación vamos a utilizar S_t en vez de $S(t)$, $\forall t > 0$.

2.1.11. Volatilidad

La volatilidad mide la incertidumbre acerca del precio futuro de un activo. En teoría, la volatilidad se calcula continuamente, para valuar las opciones, tal como se dan los cambios en el valor de S . Black-Scholes asume σ constante, esto implica una previa estimación estadística de σ , por ejemplo medir el comportamiento de los precios en los últimos meses y usar estimadores de varianza (volatilidad Histórica).

Como vimos en la sección 2.1.10 la fórmula de Black-Scholes para una opción call depende de distintos parámetros, el precio inicial del activo, la tasa libre de riesgo, el strike, la madurez de la opción y su volatilidad. Pero esta última no la brinda el mercado.

Volatilidad Histórica

La volatilidad histórica como su nombre lo indica, muestra el riesgo histórico de un período de tiempo de hoy hacia atrás. Se calcula midiendo las variaciones que han tenido los rendimientos del activo en cierto período de tiempo, que puede ser 20 días, 50 días o 200 días o el que cada analista considere mejor y esta volatilidad por lo regular se presenta anualizada. La metodología mas común para calcularla es calculando las desviaciones estándar de los rendimientos del activo. La fórmula 2.7 muestra el cálculo de la volatilidad histórica.

$$x_t = \ln \left(\frac{S_t}{S_{t-1}} \right) \quad (2.5)$$

$$\bar{X} = \frac{1}{n} \sum_{t=1}^n x_t \quad (2.6)$$

$$\text{Volatilidad Historica} = \frac{1}{\Delta t} \cdot \sqrt{\frac{1}{n-1} \sum_{t=1}^n (x_t - \bar{X})^2} \quad (2.7)$$

donde S_t es el precio del activo en el día t , n la cantidad de días sobre la cual calcularemos la volatilidad histórica, y Δt es el factor que permite dar la volatilidad histórica anualizada, en general $\Delta t = \frac{1}{252}$ ya que 252 es la cantidad de días comerciales al año.

Volatilidad Implícita

La volatilidad implícita es un factor clave en el trading de opciones que refleja las expectativas respecto al desplazamiento futuro del precio de un activo subyacente. Alta volatilidad implícita indica que los participantes del mercado esperan un movimiento mayor en el futuro y, por el contrario, baja volatilidad apunta hacia un desplazamiento posiblemente menor.

Como vimos en la sección 2.1.10 para valorar una call europea necesitamos conocer, el precio inicial del subyacente, la tasa libre de riesgo, el strike, el tiempo de madurez de la opción y la volatilidad σ del subyacente. Los primeros cuatro parámetros son conocidos al momento de iniciar la opción. En cambio σ representa una volatilidad del activo en el período de vigencia de la opción, y por lo tanto es desconocido. Más aún, se está suponiendo constante cuando en la práctica puede ser un valor variable e incluso estocástico. Dado que las opciones call cotizan en el mercado, también es conocida la prima de la opción. Por ello se denomina volatilidad implícita al valor de σ que iguala la prima en el mercado de la opción con la correspondiente fórmula de Black-Scholes.

Cuadro 2.2: Superficie de volatilidad

	K/S_0				
	0,9	0,95	1,00	1,05	1,10
1 mes	14,2	13,0	12,0	13,1	14,5
3 meses	14,0	13,0	12,0	13,1	14,2
6 meses	14,1	13,3	12,5	13,4	14,3
1 año	14,7	14,0	13,5	14,0	15,1
2 años	15,0	14,4	14,0	14,5	15,1
5 años	14,8	14,6	14,4	14,7	15,0

2.1.12. Superficie de volatilidad

Para valorar las opciones los agentes usan la superficie de volatilidad, ya que la volatilidad depende del strike y de la madurez.

Una superficie de volatilidad es una representación tridimensional de las volatilidades implícitas de un subyacente en relación con los diferentes precios de ejercicio y las diferentes fechas de madurez. La superficie de volatilidad combina las sonrisas de volatilidad [7] con la estructura temporal de la volatilidad para tabular las volatilidades adecuadas y de este modo poder valorar una opción con cualquier precio de ejercicio y fecha de vencimiento. Un ejemplo de superficie de volatilidad puede ser usada para valorar opciones sobre divisas como se muestra en el Cuadro 2.2. En este caso asumimos que la sonrisa es medida como la relación entre la volatilidad y $\frac{K}{S_0}$.

Una dimensión de la tabla es $\frac{K}{S_0}$ (segunda fila) y la otra es el tiempo de madurez (primera columna). Los valores del cuadro son las volatilidades implícitas calculadas a partir de la fórmula de Black-Scholes. Las volatilidades que no se encuentren en la tabla son calculadas mediante interpolación, por lo general se utilizan Spline cúbicos [8].

2.2. Método de Bisección

El método de bisección se basa en el Teorema de los valores intermedios [9]. Si f es continua en un intervalo cerrado $[a, b]$ y $f(a)f(b) < 0$, entonces f tiene al menos una raíz en el intervalo abierto (a, b) .

Si el algoritmo de bisección se aplica a una función continua f en un intervalo $[a, b]$, donde $f(a)f(b) < 0$, entonces, después de n pasos se habrá calculado una raíz aproximada con un error a lo más de $\frac{(b-a)}{2^{n+1}}$ [10].

El Algoritmo 2.1 detalla el método de bisección.

2.3. Método de Brent

El método de Brent es un algoritmo de búsqueda de raíces que combina el método de bisección, el método secante y la interpolación cuadrática inversa. Este método converge siempre que los valores de la función sean computables dentro de una región dada que contiene una raíz.

A diferencia del método de Bisección, el método de Brent converge haciendo menos iteraciones sin perder la robustez del método de bisección ya que para ciertos casos utiliza el método de bisección, sin caer en los problemas de divergencia que tienen el método de interpolación cuadrática inversa, o del método de la secante.

Método de la Secante

En el método de la secante imita el de Newton 2.8 pero evita el cálculo de derivadas. Entonces se reemplaza la derivada en el punto x_n , $f'(x_n)$, de la fórmula de Newton por una aproximación de su derivada 2.9 [11].

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2.8)$$

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \approx \frac{f(x+h) - f(x)}{h} \quad (2.9)$$

Luego el método de la secante esta dado por:

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (2.10)$$

Interpolación cuadrática inversa

Es un método iterativo, que aplica la fórmula de interpolación de Lagrange para hacer una interpolación cuadrática en el inverso de f . Para hallar la raíz de la ecuación se implementa el método iterativo [12]:

$$\begin{aligned} x_{n+1} = & \frac{f_{n-1}f_n}{(f_{n-2} - f_{n-1})(f_{n-2} - f_n)} x_{n-2} \\ & + \frac{f_n f_{n-2}}{(f_{n-1} - f_n)(f_{n-1} - f_{n-2})} x_{n-1} \\ & + \frac{f_{n-1} f_{n-2}}{(f_n - f_{n-1})(f_n - f_{n-2})} x_n \end{aligned}$$

donde $f_k = f(x_k)$. El algoritmo 2.2 detalla el método de Brent.

Algorithm 2.1: Bisección

Input : $f, a, b, \text{tol}, \epsilon$
Output: c
Precondition: $f(a)f(b) < 0$ and $a < b$

```

1 err := b - a;
2 while tol < err do
3   c := (a+b)/2;
4   err := err/2;
5   fc := f(c);
6   if |fc| <  $\epsilon$  then
7     return;
8   end
9   if  $f(a)fc < 0$  then
10    b := c;
11  else
12    a := c;
13  end
14 end
15 return;
```

2.4. Redes neuronales

Una Red Neuronal Artificial es un modelo matemático inspirado en el comportamiento biológico de las neuronas y en la estructura del cerebro, y que es utilizada para resolver un amplio rango de problemas. Su diseño consiste en un conjunto de variables de entrada y de salida con ciertas funciones que relacionan unas con otras y que están determinadas por ciertos parámetros. La red neuronal se entrena para estimar los parámetros óptimos (W y b) para una función $f(x, W, b)$, a partir de un conjunto de datos de entrenamiento $\{(x_1, y_1), \dots, (x_n, y_n)\}$, con el objetivo de lograr que $f(x_i, W, b) \approx y_i$, donde y_i es la salida esperada para el dato de entrada x_i para cada $i = 1, \dots, n$. Una vez entrenada la red neuronal ésta es utilizada para computar $f(x, W, b)$ para otros datos x de interés.

2.4.1. Perceptrón

La unidad básica de una red neuronal son los perceptrones (o neuronas). Un perceptrón toma un vector de entradas de valor real, calcula la combinación lineal de estas entradas con respecto a los pesos, y luego genera una salida, dependiendo de la función de activación. Más precisamente, la salida o para un vector de entradas (x_1, \dots, x_n) es de la forma:

$$o(x_1, \dots, x_n) = \psi(w_0 + x_1w_1 + \dots + x_nw_n),$$

Algorithm 2.2: Brent

Input : $f, a, b, \text{tol}, \epsilon$
Output: b
Precondition: $f(a)f(b) < 0$

```

1 if  $|f(a)| < |f(b)|$  then
2   | swap(a,b);
3 end
4  $c := a$ ;
5  $mflag := \text{True}$ ;
6 while  $\text{tol} < |b - a|$  or  $|f(b)| < \epsilon$  do
7   | if  $|f(a) - f(c)| > \epsilon$  and  $|f(b) - f(c)| > \epsilon$  then
8     |  $A := \frac{af(b)f(c)}{(f(a) - f(b))(f(a) - f(c))}$ ;
9     |  $B := \frac{bf(a)f(c)}{(f(b) - f(a))(f(b) - f(c))}$ ;
10    |  $C := \frac{cf(a)f(b)}{(f(c) - f(a))(f(c) - f(b))}$ ;
11    |  $s := A+B+C$ ; (interpolacion cuadrática inversa)
12  | else
13    |  $s := b - f(b) \frac{b - a}{f(b) - f(a)}$ ; (metodo de secante)
14  | end
15  | if not  $\left(\frac{3a + b}{4} < s < b\right)$  or ( $mflag$  and  $|s - b| \geq |b - c|/2$ ) or (not
    |  $mflag$  and  $|s - b| \geq |c - d|/2$ ) or ( $mflag$  and  $|b - c| < \text{tol}$ ) or (not  $mflag$ 
    | and  $|c - d| < \text{tol}$ ) then
16    |  $mflag := \text{True}$ ;
17    |  $s := \frac{a + b}{2}$ ; (metodo de bisección)
18  | else
19    |  $mflag := \text{False}$ ;
20  | end
21  |  $d := c$ ;
22  |  $c := b$ ;
23  | if  $f(a)f(s) < 0$  then
24    |  $b := s$ 
25  | else
26    |  $a := s$ 
27  | end
28  | if  $|f(a)| < |f(b)|$  then
29    | swap(a,b);
30  | end
31 end
32 return;

```

donde cada w_i es una constante de valor real, o peso, y ψ es la función de activación [13]. La Figura 2.7 ilustra un perceptrón:

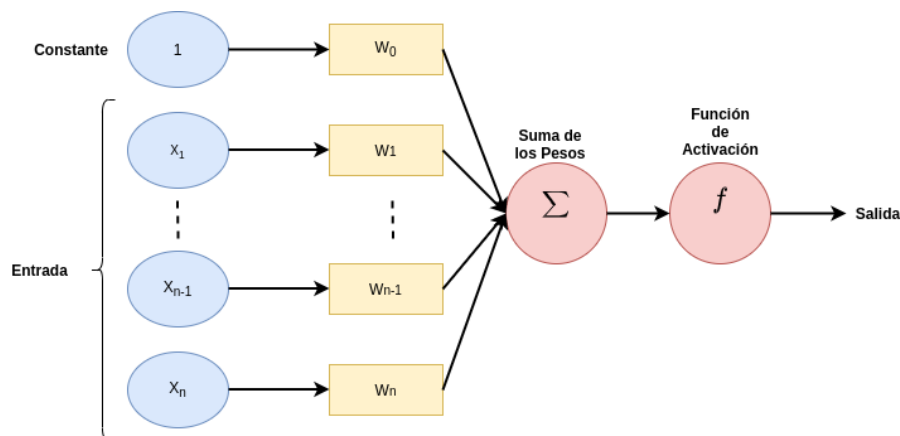


Figura 2.7: Esquema del perceptrón

2.4.2. Redes Feed-Forward

Este tipo de algoritmo recibe el nombre de “red” porque se construye componiendo funciones (perceptrones).

La arquitectura de este tipo de modelos se puede dividir en tres partes principales, denominadas capas, tal como se muestra en la Figura 2.8. La característica distintiva de este tipo de red es que no existen conexiones entre neuronas de una misma capa. Las capas a su vez están completamente conectadas (o fully connected) debido a que cada neurona de una capa se conecta con todas las neuronas en la capa siguiente.

En la Figura 2.8 podemos observar las distintas capas que componen una red neuronal. En primer lugar tenemos la capa de entrada, donde se define cuántos valores de entrada tomará nuestro modelo, estos serán luego enviados a la primera capa oculta. Después de esto puede venir una o más capas ocultas, seguidas de una capa de salida que genera una aproximación del resultado deseado, por ende, si una red neuronal Feed-Forward tiene N capas, entonces tiene $N - 2$ capas ocultas. Cada unidad (o input) alimenta solo las unidades de la siguiente capa. Las unidades en intermedio de las capas a menudo se llaman capas ocultas (hidden units) porque no tienen conexión directa con el conjunto de datos, tanto de entrada como de salida [14].

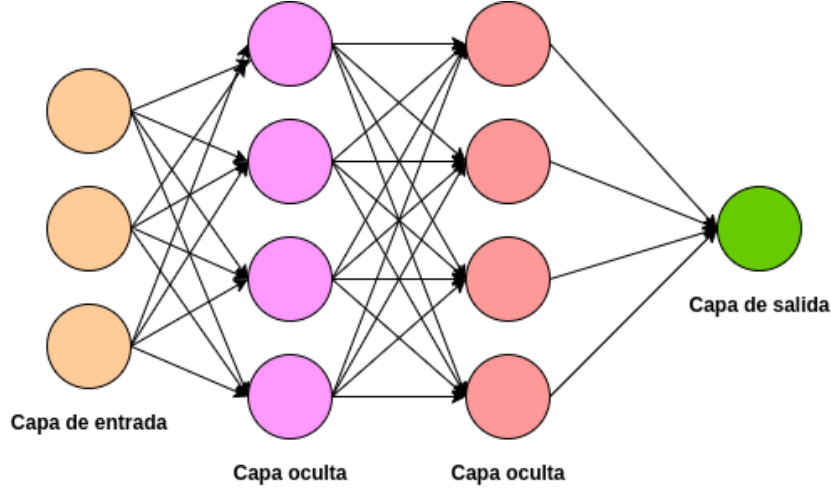


Figura 2.8: Feed-Forward

2.4.3. Aprendizaje por Descenso por el Gradiente

Como hemos mencionado anteriormente, el objetivo de una red neuronal es estimar los parámetros W y b óptimos para una función $f(x, W, b)$. Una forma de estimar esos parámetros es reducir el error medido por una función de costo.

$$J(x, W, b, Y) = \frac{1}{n} \sum_{i=1}^n (f(x_i, W, b) - y_i)^2 \quad (2.11)$$

donde $Y = (y_1, \dots, y_n)$, $x = (x_1, \dots, x_n)$.

Para reducir dicho error se buscan W y b tal que se minimice la función $J(x, W, b, Y)$. Para esto se puede utilizar el método de descenso por el gradiente.

Para simplificar la notación escribimos $w = [W, b]$, con $w_0 = b$. El gradiente de J en w , denotado como $\nabla J(w)$, es el vector de derivadas parciales de J , es decir, $\nabla J(w) = \left(\frac{\partial J(w)}{\partial w_0}, \frac{\partial J(w)}{\partial w_1}, \frac{\partial J(w)}{\partial w_2}, \dots, \frac{\partial J(w)}{\partial w_n} \right)$.

El descenso por el gradiente es un algoritmo iterativo, se comienza con valor inicial de w^0 (por ejemplo $w^0 = [0, 0, \dots, 0]$) y entonces en cada iteración (época) damos un paso en la dirección negativa del gradiente en el punto actual. Esto es, $w^{t+1} = w^t - \eta \nabla J(w^t)$, donde $\eta > 0$, conocido como la tasa de aprendizaje, es el encargado de determinar la longitud del paso. Ya que $\nabla J(x, w^t, Y, b) > 0$ cuando J es creciente en w^t , y $\nabla J(x, w^t, Y, b) < 0$ cuando J es decreciente en w^t (como muestra la Figura 2.9), se obtiene $J(x, w^{t+1}, b, Y) \leq J(x, w^t, b, Y)$, siempre y cuando η no sea muy grande (Ver Figura 2.9).

El Algoritmo 2.3 aplica el método del descenso por el gradiente.

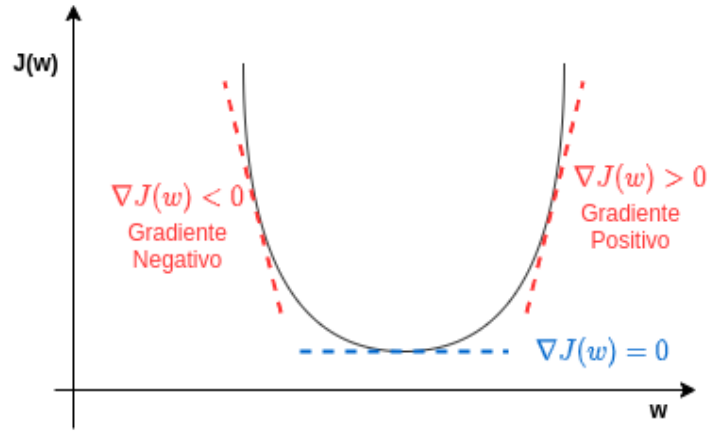


Figura 2.9: Representación esquemática del descenso por el gradiente

Algorithm 2.3: Representación esquemática del descenso por el gradiente

Input : x, Y, η, T
Output: w
Precondition: $d = \text{len}(w)$

```

1 initialize  $w^0$ ;
2  $t := 0$ ;
3 while  $t < T$  do
4    $i := 0$ ;
5   while  $i < d$  do
6      $w_i^{t+1} = w_i^t - \eta \nabla J(x, w_i^t, Y, b)$ ;
7      $i := i+1$ ;
8   end
9    $t := t+1$ ;
10 end
11 return;

```

2.4.4. Funciones de Activación

La función de activación en las redes neuronales son las encargadas de enviar señales al siguiente nivel o siguiente capa. Es crucial elegir la función de activación correcta para no caer en el *problema de desvanecimiento de gradiente*.

Otra importante característica que debe tener la función de activación es ser diferenciable, ya que al aplicar el algoritmo del *descenso por el gradiente*, cuando se propaga para hacia atrás, se calculan los gradientes de error con respecto a los pesos para calcular los nuevos pesos acordemente [15].

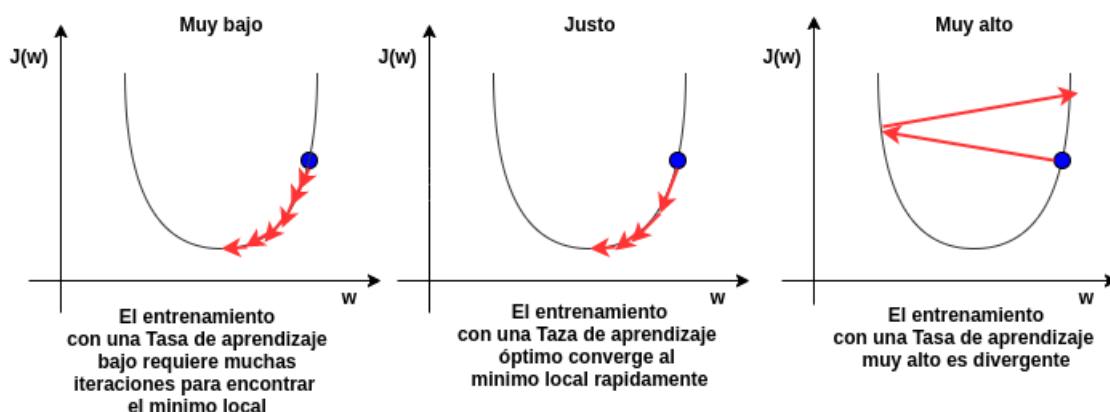


Figura 2.10: Efecto de la tasa de aprendizaje sobre el entrenamiento

Las fórmulas 2.12, 2.13, 2.14 muestra las funciones de activación que vamos a utilizar, en nuestro caso el α de la Elu es 1.

$$ReLU(z) = \max(z, 0) \quad (2.12)$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.13)$$

$$Elu(z) = \begin{cases} \alpha(e^z - 1) & \text{si } z < 0 \\ z & \text{si } z \geq 0 \end{cases} \quad (2.14)$$

2.4.5. Tasa de Aprendizaje

La tasa de aprendizaje determina el tamaño del paso en cada iteración mientras se mueve hacia un mínimo de una función de error. Una tasa de aprendizaje grande conduce rápidamente a un mínimo local, pero con riesgo de diverger. En cambio una tasa de aprendizaje pequeña no diverge pero necesita muchas iteraciones para llegar al mínimo local (Figura 2.10). Comúnmente se utiliza una tasa de aprendizaje grande y se la va decrementando por cada época o cada varias épocas hasta encontrar un buen resultado.

Decrecimiento de la tasa de aprendizaje

El decrecimiento de la tasa de aprendizaje es una método para entrenar redes neuronales, se empieza con una tasa de aprendizaje grande y luego se la va disminuyendo, empíricamente se obtienen mejores resultados que entrenar la red con una tasa de aprendizaje constante. Las fórmulas 2.15, 2.16 y 2.17 describen algunos métodos de decrecimiento:

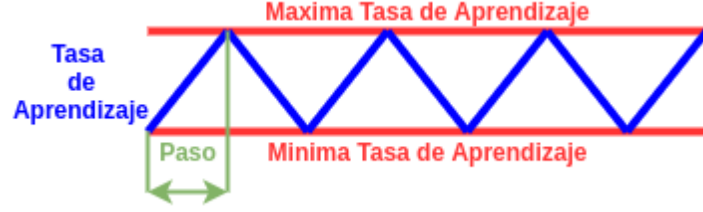


Figura 2.11: Cyclical Learning Rate

$$TimeBasedDecay(epoch) = base_lr \frac{1}{1 + decay \cdot epoch} \quad [16] \quad (2.15)$$

$$StepDecay(epoch) = base_lr \cdot decay \left\lfloor \frac{1 + epoch}{epoch_drop} \right\rfloor \quad [16] \quad (2.16)$$

$$ExponentialDecay(epoch) = base_lr \cdot e^{(-k \cdot epoch)} \quad [17] \quad (2.17)$$

donde $\lfloor x \rfloor$ indica la parte entera inferior de x , $epoch$ representa la época en cual la red se encuentra, $base_lr$ es el valor inicial de la tasa de aprendizaje, $decay$ es el coeficiente de decrecimiento de la tasa de aprendizaje y $epoch_drop$ cada cuantas épocas se decrece la tasa de aprendizaje (Solo en el Step Decay).

Cyclical Learning Rate

En este caso la tasa de aprendizaje varía entre un mínimo y máximo, creciendo y decreciendo como muestra la Figura 2.11. Para obtener dicho máximo y mínimo utilizaremos un método similar al de Smith [18]. A diferencia del método de Smith, iremos subiendo la tasa de aprendizaje exponencialmente por cada iteración de la red. Empezaremos con una tasa de aprendizaje de 10^{-10} hasta llegar a 1, comparando la tasa de error contra la tasa de aprendizaje.

2.5. k -fold cross validation

k -fold cross validation es una técnica utilizada para evaluar modelos propuestos, con el fin de encontrar el mejor modelo. En k -foldcross validation los datos de prueba se dividen en k subconjuntos. Uno de los subconjuntos se utiliza como datos de test y el resto $(k - 1)$ como datos de entrenamiento. El proceso de cross validation es repetido durante k iteraciones, con cada uno de los posibles subconjuntos de datos de test. Finalmente se realiza el promedio de los resultados de cada modelo(cada modelo tiene k resultados),

guardando el modelo que obtuvo mejor promedio. Este método es muy preciso puesto que evaluamos a partir de k combinaciones de datos de entrenamiento y de test.

Capítulo 3

Implicancias de la volatilidad implícita

En esta sección describiremos diferentes usos de la volatilidad implícita. El primer uso es en el cálculo de superficies de volatilidad, la cual sirve para valorar otros tipos de opciones. El segundo uso es utilizar la volatilidad implícita como indicador de riesgo de mercado, de esta manera poder ejercer estrategias, ya sean de cobertura o de especulación.

3.1. Superficie de Volatilidad para valorar opciones

Para valorar las opciones europeas, lookback, asiáticas, entre otras, existen modelos matemáticos para calcular el valor la opción. A continuación listaremos una serie de modelos desarrollados para valorar distintos tipos de opciones vainilla y exóticas [19].

- Black-Scholes(1973): Modelo para valorar opciones europeas.
- Rubinstein (1991): Modelo para valorar opciones chooser simples.
- Conze y Viswanathan(1991): Modelo para valorar opciones lookbacks con precio de ejercicio fijo.
- Goldman, Sosin y Gatto(1979): Modelo para valorar opciones lookbacks con precio de ejercicio flotante.
- Kemma y Vorst (1990): Modelo para valorar opciones asiáticas con media geométrica.
- Levy (1992): Modelo para valorar opciones asiáticas con media aritmética.
- Margrabe (1978): Modelo para valorar opciones sobre el intercambio de dos activos.

Todos los modelos anteriormente mencionados para ser aplicados necesitan la volatilidad entre otros parámetros. Pero si el mercado no nos brinda el valor de la volatilidad hay que estimarla, una manera es mediante la volatilidad implícita, pero como la volatilidad depende del strike y la madurez, pueden haber casos en que no puedo obtener la volatilidad para algún contrato, entonces en esos casos se estima la volatilidad con superficie de volatilidad visto en el capítulo anterior.

3.2. Estrategias dependiendo de la volatilidad implícita

Entender el concepto de volatilidad es esencial para tener éxito en la comercialización de opciones. Un inversor que puede reconocer cuándo una opción o series de opciones están baratas o caras tiene una gran ventaja a la hora de invertir. Esto quiere decir que si la volatilidad implícita es baja, entonces el prima de la opción va a ser baja. En cambio si la volatilidad implícita es alta, el prima de la opción va a ser alta. Ya que el prima de la opción es creciente con respecto a la volatilidad [4].

Ahora bien, ¿cómo reconocer cuándo la volatilidad implícita es alta o baja?. Una manera es comparar las volatilidades históricas anuales calculadas mensualmente de la opción sobre los últimos 2 años contra la volatilidad implícita de la opción. Por ejemplo si la volatilidad implícita es aproximadamente menor al 70 % de las volatilidades históricas anuales calculadas mensuales de los últimos 2 años (hay 24 volatilidades históricas calculadas), entonces la volatilidad implícita es baja.

Ahora nombraremos algunas estrategias en las cuales se hace uso de la volatilidad implícita [20].

Terminología. En las siguientes definiciones consideremos una opción europea con strike K sobre un subyacente con valor $S(0)$.

- ATM (at-the-money): Una opción está ATM cuando $S(0) = K$ tanto en opciones put como call.
- OTM (out-of-the-money): Una opción call está OTM cuando $S(0) < K$ y una opción put cuando $S(0) > K$. Es decir que si se ejerce la opción en ese momento el payoff es cero.
- ITM (at-the-money): Una opción call está ITM cuando $S(0) > K$ y en una opción put cuando $S(0) < K$. Es decir que si se ejerce la opción en ese momento el payoff es $|S(0) - K|$.

3.2.1. Backspread

Las claves para aplicar la estrategia Backspread son las siguientes:

- El inversor espera un movimiento particular del mercado, pero teniendo un poco de protección en caso de equivocarse.
- Busca que la volatilidad implícita sea baja al momento de aplicar la estrategia, con esperanza que suba en el futuro.

La estrategia Backspread involucra suscribir opciones call o put, ATM o ITM, las cuales tienen primas caras ya que el inversor que compra la opción tiene más posibilidades de obtener un payoff positivo y entonces es más probable que la opción sea ejecutada, y simultáneamente comprar un número mayor de opciones OTM, las cuales tienen primas baratas porque es más difícil que el inversor que compra la opción ejerza. Así, si se suscriben n opciones con primas s_1, s_2, \dots, s_n , y se compran m opciones con primas c_1, c_2, \dots, c_m respectivamente, idealmente se busca la relación:

$$\sum_{i=1}^n s_i > \sum_{j=1}^m c_j \quad (3.1)$$

En el caso que tengamos expectativas de un movimiento alcista del subyacente entonces la estrategia será Call Backspread, que consiste en vender y comprar opciones call. El objetivo es que el precio del subyacente suba o baje drásticamente para obtener ganancia, con preferencia alcista como se puede observar en la Figura 3.1, en el caso que se suscriba una opción y se compren dos su payoff está dado por:

$$\text{Payoff} = \min(K_1 - S_T, 0) + 2 \max(S_T - K_2, 0), \quad (3.2)$$

y su ganancia por:

$$\text{Ganancia} = \min(K_1 - S_T + c_1, c_1) + 2 \max(S_T - K_2 - c_2, -c_2) \quad (3.3)$$

donde c_1 es la prima de la opción suscrita con strike K_1 , c_2 es la prima de la opción comprada con strike K_2 , y ambas opciones sobre un subyacente con valor S_T el valor del subyacente en el tiempo T .

En el caso que tengamos expectativas de un movimiento bajista del subyacente entonces la estrategia será Put Backspread, que consiste en vender y comprar opciones put. El objetivo es el mismo que el Call Backspread, con preferencia bajista como se puede observar en la Figura 3.2, en el caso que se suscriba una opción con strike K_1 y prima p_1 y se compren dos opciones con strike K_2 de prima p_2 cada una, su payoff está dado por:

$$\text{Payoff} = \min(S_T - K_1, 0) + 2 \max(K_2 - S_T, 0) \quad (3.4)$$

y su ganancia por:

$$\text{Ganancia} = \min(S_T - K_1 + p_1, p_1) + 2 \max(K_2 - S_T - p_2, -p_2) \quad (3.5)$$

Ver Figuras 3.1 y 3.2.

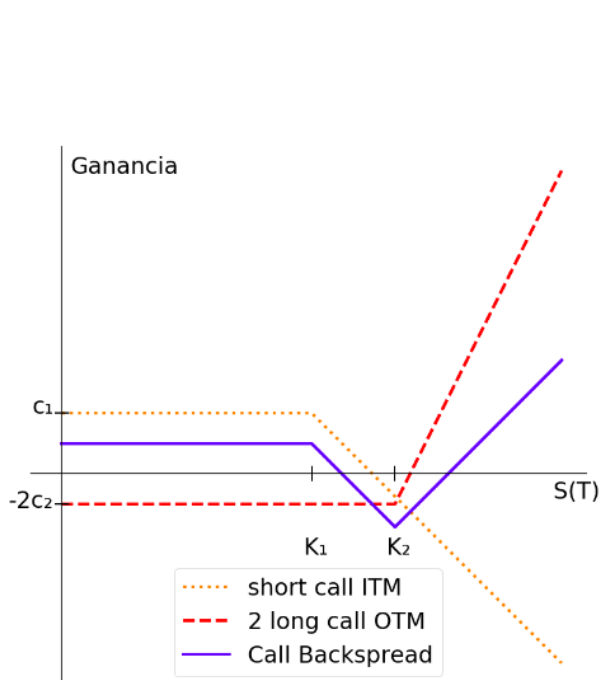


Figura 3.1: Diagrama de ganancia para una Call Backspread con madurez T .

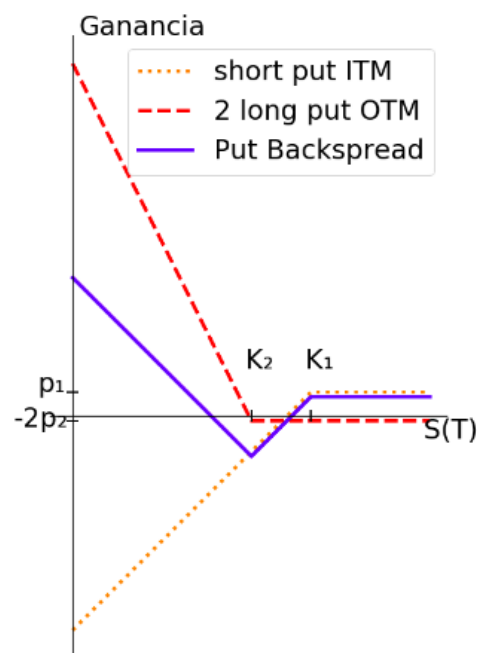


Figura 3.2: Diagrama de ganancia para una Put Backspread con madurez T .

3.2.2. Long Straddle y Long Strangle

Se denomina Long Straddle o Long Strangle porque el inversor se encuentra en posición long sobre las opciones de la estrategia.

Las claves para aplicar estas estrategias son las siguientes:

- El inversor tiene expectativa que el valor del subyacente tenga un gran cambio, sin importar si el precio sube o baja.
- Se busca que la volatilidad implícita sea baja al momento de aplicar la estrategia.
- Idealmente se busca opciones out-of-the-money.
- Se busca que sean opciones largas.

La estrategia de comprar una Straddle o una Strangle consiste en comprar una opción call y una opción put al mismo tiempo, con el mismo tiempo de madurez, ambas con el igual strike (Straddle) o diferente strike (Strangle). Idealmente se busca que al momento de comprar las opciones, la volatilidad implícita sea baja (comprar opciones baratas), y que

el precio del subyacente tenga un abrupto cambio (Por ejemplo por situaciones atípicas como la Guerra comercial entre Estados Unidos y China, Pandemia, Precio Petróleo, entre otros) sin importar su dirección, puede ser alcista o bajista. Por eso el tiempo de madurez es clave en este tipo de estrategias. La pérdida en este tipo de estrategias es limitada, es el valor de las primas. En cambio la ganancia puede ser ilimitada.

El payoff y ganancia para una Straddle o Strangle que consiste en comprar una opción put con strike K_1 y prima p y comprar una opción call con strike K_2 y prima c :

$$\text{Payoff} = \max(S_T - K_2, 0) + \max(K_1 - S_T, 0) \quad (3.6)$$

$$\text{Ganancia} = \max(S_T - K_2 - c, -c) + \max(K_1 - S_T - p, -p) \quad (3.7)$$

En el caso de una estrategia straddle, donde $K_1 = K_2 = K$, las fórmulas para el payoff y la ganancia se traducen en:

$$\text{Payoff} = |S_T - K| \quad (3.8)$$

$$\text{Ganancia} = |S_T - K| - c - p. \quad (3.9)$$

Los diagramas de ganancia de las estrategias strangle y straddle se ilustran en las Figuras 3.3 y 3.4.

3.2.3. Long Butterfly Spread

Las claves para aplicar la estrategia son los siguientes:

- El inversor tiene expectativa que el valor del subyacente no varíe mucho hasta su vencimiento.
- Se busca que la volatilidad implícita sea alta al momento de aplicar la estrategia, cuánto más alta mejor.
- Se invierte en opciones cortas, con un tiempo de expiración menor a 60 días.

La estrategia Long Butterfly Spread (Long significa que el valor de las primas en posición long son mayores a las primas en posición short) puede utilizarse tanto con opciones call, como con opciones put.

En caso que se utilicen opciones call, la estrategia consiste en comprar una opción call con un determinado strike K_1 , vender dos call con un strike mayor al strike de la opción nombrada anteriormente, K_2 , y comprar otra opción call con un strike mayor a todos los anteriores, K_3 . Esto es, $K_1 < K_2 < K_3$.

En caso que nuestra estrategia utilice opciones put, la estrategia consiste en comprar una opción put a un determinado strike, vender dos put con un strike menor al strike

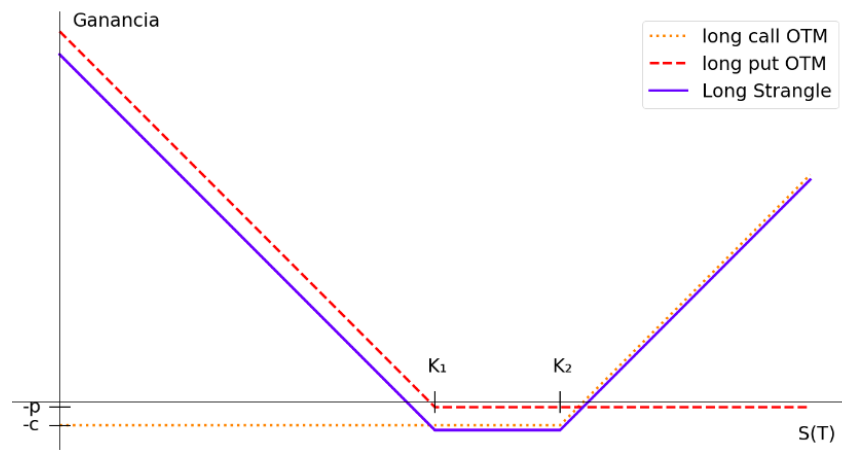


Figura 3.3: Diagrama de ganancia para una Long Strangle con madurez T .

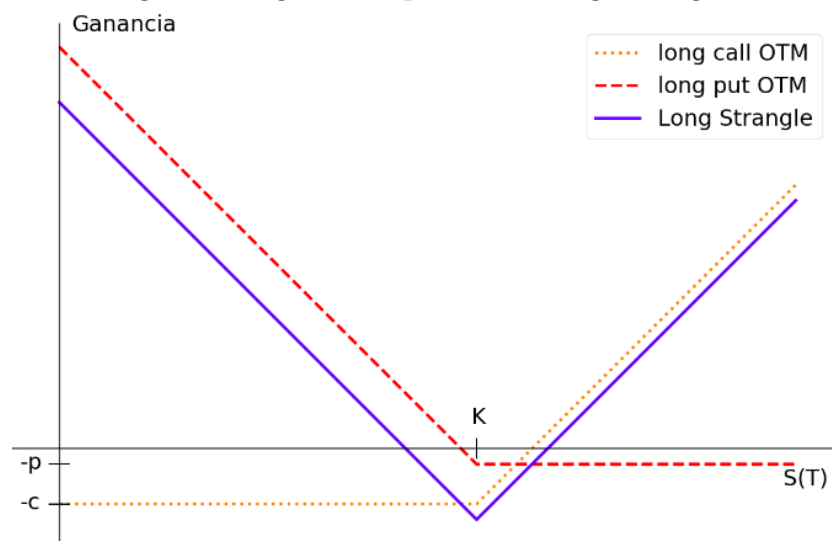


Figura 3.4: Diagrama de ganancia para una Long Straddle con madurez T .

de la opción nombrada anteriormente, y comprar otra opción put con un strike menor a todos los anteriores.

En esta estrategia la mayor ganancia se obtiene cuando el precio del subyacente se encuentre próximo al strike del medio (opciones vendidas). Pero al ser la volatilidad alta, el precio del subyacente es muy variable, siendo así esta estrategia riesgosa. Una buena medida es abandonar la estrategia cuando se haya obtenido una buena ganancia, sin esperar que el precio del subyacente llegue al pico, como se observa en la Figura 3.5. Una estrategia en la que se compra dos opciones call con strike K_1 y K_3 de primas c_1 y c_3 respectivamente y se suscriben 2 opciones call con strike K_2 y prima c_2 tendrá un payoff y una ganancia dada por:

$$\text{Payoff} = \max(S_T - K_1, 0) + 2 \min(K_2 - S_T, 0) + \max(S_T - K_3, 0) \quad (3.10)$$

$$\text{Ganancia} = \max(S_T - K_1 - c_1, -c_1) + 2 \min(K_2 - S_T + c_2, c_2) + \max(S_T - K_3 - c_3, -c_3) \quad (3.11)$$

donde S_T es el valor del subyacente en el tiempo T .

El diagramas de ganancia de la estrategia Long Butterfly Spread se ilustran en las Figuras 3.5

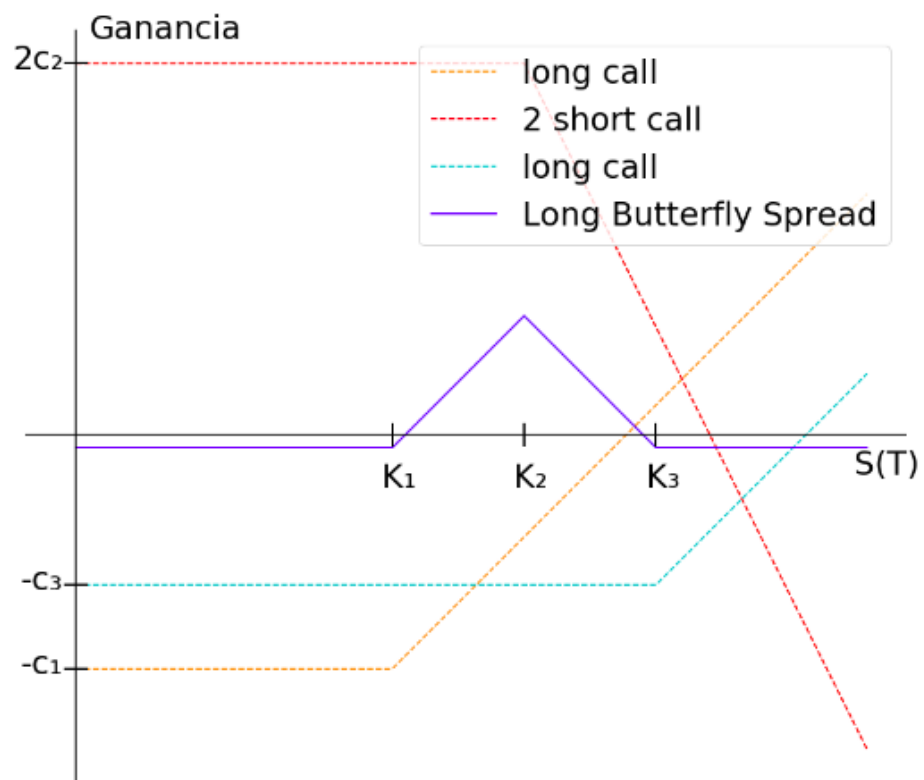


Figura 3.5: Diagrama de ganancia para una Long Butterfly Spread con madurez T .

Capítulo 4

Implementación numérica del método de Bisección y de Brent

En este capítulo comentaremos detalles de la implementación del método de Bisección y del método Brent para aproximar la volatilidad implícita a partir de la fórmula de Black-Scholes.

4.1. Introducción

Tanto el método de Bisección como el método de Brent son algoritmos de búsqueda de raíces de funciones, como lo es nuestro problema de encontrar el valor de la volatilidad implícita a partir de la fórmula de Black-Scholes (2.1).

Sea c la prima de una opción call europea con madurez T y strike K sobre una acción cuyo precio actual es $S(0)$, siendo r la tasa de interés libre de riesgo. La función a la cual aplicaremos ambos métodos numéricos está dada por la siguiente fórmula:

$$g(\hat{\sigma}) = S(0) \Phi(d_1(\hat{\sigma})) - K e^{-rT} \Phi(d_2(\hat{\sigma})) - c \quad (4.1)$$

donde d_1 es la función 2.3, d_2 es la función 2.4, Φ es la función de distribución normal estándar acumulada 2.2 y c es la prima de una opción call europea.

Se puede probar que g es monótona creciente. Para esto, si definimos:

$$B(\sigma) = S(0)\Phi(d_1(\sigma)) - K e^{-rT}\Phi(d_2(\sigma)) \quad (4.2)$$

donde d_1 es la función 2.3 y d_2 es la función 2.4,

es:

$$B'(\sigma) = K e^{-rT} \Phi'(d_2(\sigma)) \sqrt{T} \quad (4.3)$$

y esta expresión es mayor a 0 para todo $\sigma > 0$. Luego como B es una función monótona creciente con respecto a σ , y además se tiene que

$$g(\sigma) = B(\sigma) - c \quad (4.4)$$

concluimos que g también es creciente con respecto a σ , ya que c es constante.

4.2. Implementación de los métodos numéricos

En esta sección se implementará el método de bisección y de Brent para el cálculo de la volatilidad implícita utilizando la fórmula de Black-Scholes. Para cualquiera de los métodos se denotará a_0 como el extremo inferior del intervalo al cual se le aplicará el método numérico, b_0 como el extremo superior del intervalo al cual se le aplicará el método numérico, a_n y b_n como los extremos del intervalo en la n -ésima iteración del método numérico, y ε como la tolerancia del método numérico.

4.2.1. Intervalo inicial para el método de bisección y de Brent

Como mostramos de la sección anterior, $g : (0, \infty) \rightarrow \mathbb{R}$ es una función monótona creciente. Entonces podemos tomar a_0 lo suficientemente pequeño para que $g(a_0) < 0$, y $b_0 > a_0$ que cumpla $g(b_0) > 0$. Por lo tanto, por el Teorema de los valores intermedios y porque g que es estrictamente creciente, existe un único $\hat{\sigma} \in [a_0, b_0]$ (intervalo inicial) tal que $g(\hat{\sigma}) = 0$. El siguiente algoritmo determina b_0 .

Algorithm 4.1: Encontrar b

Input : g

Output: b

```

1  $b := 1$ ;
2 while  $g(b) < 0$  do
3   |  $b := b * 10$ ;
4 end
5 return;
```

4.2.2. Aplicación de los Métodos Numéricos

Luego para aplicar los métodos numéricos inicializamos el intervalo $[a_0, b_0]$ de manera que $a_0 < b_0$ y $g(a_0) < 0 < g(b_0)$. Luego aplicamos el algoritmo de bisección 2.1 o Brent 2.2 hasta hallar un ξ , tal que $g(\xi) = 0$ ó $|b_n - a_n| < \varepsilon$, con $n > 0$.

Capítulo 5

Implementación con Red Neuronal Feed-Forward

En este capítulo presentamos el cálculo de la volatilidad implícita mediante el diseño y uso de redes neuronales.

5.1. Cálculo de prima de opción call usando S/K

Rene Garcia y Ramazan Gençay [2], invocando la homogeneidad de la fórmula de Black-Scholes, demostraron que las redes neuronales estiman mejor el precio de una opción usando el cociente S/K . Así podemos reescribir la fórmula de Black-Scholes 2.1, obteniendo 5.2.

$$\begin{aligned}\frac{c}{K} = \frac{B(S(0), K, r, \sigma, T)}{K} &= \frac{S(0)}{K} \Phi \left(d_1 \left(\frac{S(0)}{K} \right) \right) - e^{-rT} \Phi \left(d_2 \left(\frac{S(0)}{K} \right) \right) \quad (5.1) \\ &= \tilde{B} \left(\frac{S(0)}{K}, T, r, \sigma \right)\end{aligned}$$

donde d_1 es la ecuación 2.3 y d_2 es la ecuación 2.4.

5.2. Generación de muestra

Como hemos visto en la sección anterior, la fórmula (5.2) nos da un valor para $\frac{c}{K}$. En nuestro caso buscamos estimar la volatilidad implícita, pero usaremos el cociente para generar la muestra ya que en la práctica la red estima mejor, observar que podemos generar la muestra del tamaño que deseemos, por lo general generaremos muestras de 10^6 elementos para entrenamiento, ya que consideramos que es una muestra lo suficientemente

grande para no tener problemas de sobreentrenamiento sin tener tanto costo en el tiempo de entrenamiento, en tanto las muestras de validación y test serán de 10^5 elementos.

Luego vamos a generar dos muestras, una muestra amplia y una muestra estrecha. La muestra amplia va a ser para el entrenamiento de la red, y la muestra angosta para la evaluación de la red.

Las variables de la muestra ya sea amplia o estrecha, pertenecerán a un ambiente previamente definido, pero la muestra estrecha será definida sobre un ambiente mas pequeño que el ambiente de la muestra amplia, ya que la red estima peor en los extremos, como puede verse en la Cuadro 5.2.

	Parámetros	muestra amplia	muestra estrecha
Entrada	precio ratio(S_0/K)	$[M^1, N^1]$	$[M^1 + D_1^1, N^1 - D_2^1]$
	Tiempo de madurez(τ)	$[M^2, N^2]$	$[M^2 + D_1^2, N^2 - D_2^2]$
	volatilidad(σ)	$[M^3, N^3]$	$[M^3 + D_1^3, N^3 - D_2^3]$
	Tasa libre de riesgo(r)	$[M^4, N^4]$	$[M^4 + D_1^4, N^4 - D_2^4]$
Salida	Prima de Call(c/K)	(O^1, L^1)	(O^2, L^2)

Cuadro 5.1: Rango de los parámetros para la generación de las muestras $M^i + D_1^i < N^i - D_2^i$, $D_j^i > 0$, $O^j < L^j$, para $i = 1, 2, 3, 4$, $j = 1, 2$

Luego para generar una muestra de tamaño N , aplicaremos N veces el algoritmo 5.1, donde:

- **número_aleatorio:** Es una función que dado un rango genera un número aleatorio perteneciente a dicho rango.
- **\tilde{B} :** Es la fórmula de Black-Scholes 5.2 dada en la sección anterior.
- **rango_ratio:** Es el rango de valores del ratio(S_0/K).
- **rango_T:** es el rango de valores de el tiempo de madurez(τ).
- **rango_σ:** es el rango de valores de la volatilidad implícita (σ).
- **rango_r** es el rango de valores de la tasa libre riesgo(r).
- C es c/K

Algorithm 5.1: Generación muestra

Input : rango_ratio, rango_T, rango_σ, rango_r**Output:** C

```

1 ratio := numero_aleatorio(rango_ratio);
2 T := numero_aleatorio(rango_T);
3 σ := numero_aleatorio(rango_ratio);
4 r := numero_aleatorio(rango_r);
5 C :=  $\tilde{\mathbf{B}}$ (ratio, T, σ, r);
6 return;

```

5.3. *k*-fold cross validation

En nuestro caso utilizaremos 8-fold cross validation porque nos garantizará alta varianza y bajo sesgo para estimar los hiperparámetros que definen la red neuronal que queremos construir. Este método lo aplicaremos sobre una muestra de 10^4 elementos. Los hiperparámetros propuestos para aplicar 8-fold cross validation están definidos en el Cuadro 5.3. A tales fines, primero definiremos la estructura de la red neuronal entre 1 y 10 capas ocultas, variando entre 50 a 1000 neuronas por capa oculta. Luego elegiremos la función de activación (Keras [23]), la inicialización de pesos de la red (Keras [24]) y el algoritmo de optimización del método de descenso por el gradiente (Keras [25]) en simultáneo (haciendo todas las combinaciones posibles), siguiendo por determinar la mejor función de error (Keras [26]). Luego determinaremos el dropout (porcentaje neuronas que se ignoran durante el entrenamiento) y por último el tamaño del batch (cantidad de elementos de la muestra que se propagarán a través de la red).

Para iniciar la búsqueda de los hiperparámetros óptimos, se usarán los hiperparámetros por defecto que utiliza Keras Sequential [27], excepto el tamaño del batch que será de 1024, tal como se observa en la Tabla 5.2.

Las fórmulas (5.2) a (5.6) corresponden a métricas que miden el error de modelos. Las siglas corresponden a error cuadrático medio (ECM), error absoluto medio (EAM) y error porcentual absoluto medio (EPAM). En nuestro caso se utilizarán para medir la efectividad o el error de los modelos propuestos, ya sea modelos numéricos o redes neuronales.

Cuadro 5.2: Hiperparámetros por defecto

Parámetros	Opciones
Función de error	ECM
Función de activación	ReLU
Inicialización de pesos	glorot_uniform
Algoritmo de optimización	SGD
Dropout	0
Tamaño de batch	1024
Tasa de aprendizaje	0.001
Épocas	200

Cuadro 5.3: Estimación de hiperparámetros

Parámetros	Opciones o Rango
Capas	[1, 10]
Neuronas	[50, 1000]
Función de error	ECM, EAM, EPAM
Función de activación	ReLU, Elu, tanh
Inicialización de pesos	uniform, glorot_uniform, he_uniform
Algoritmo de optimización	SGD, RMSprop, Adam
Dropout	[0, 0, 2]
Tamaño de batch	[256, 2048]

$$ECM = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5.2)$$

$$EAM = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (5.3)$$

$$EPAM = 100 \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{y_i} \quad (5.4)$$

$$SS_{tot} = \sum_{i=1}^N (y_i - \bar{y})^2, \quad \text{con } \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (5.5)$$

$$R^2 = 1 - \frac{ECM}{SS_{tot}} \quad (5.6)$$

donde y_i es el valor esperado de salida de la red, \hat{y}_i es el valor que predice la red y N es

Cuadro 5.4: Hiperparámetros óptimos

Parámetros	Opciones
Capas	3
Neuronas	950
Función de error	ECM
Función de activación	ReLu
Inicialización de pesos	random_uniform
Algoritmo de optimización	Adam
Dropout	0
Tamaño de batch	1024

el tamaño de muestra.

En Cuadro 5.4 se observan los hiperparámetros obtenidos al aplicar 8-fold cross validation, que se utilizará en el entrenamiento de la red.

Tasa de Aprendizaje

En la búsqueda de los hiperparámetros óptimos (Cuadro 5.4) hemos utilizado una **tasa de aprendizaje** fija igual a 10^{-3} .

Tomando los hiperparámetros del Cuadro 5.4 utilizaremos un método similar al de Smith [18] para determinar la tasa de aprendizaje. A diferencia del método de Smith, iremos incrementando la tasa de aprendizaje exponencialmente por cada batch, la red se entrenará con una tasa de aprendizaje de 10^{-10} hasta llegar a 1 calculando su ECM. Como se puede observar en la Figura 5.1, el rango de tasa de aprendizaje óptimo se encuentra entre 5×10^{-6} y 5×10^{-3} .

Ahora proponemos tres métodos de decrecimiento del tasa de aprendizaje. Utilizaremos grid-search [28] (utilizando 90 % de la muestra para entrenamiento, 10 % para validación) para encontrar los parámetros óptimos de los algoritmos de decrecimiento del tasa de aprendizaje. Tomando los hiperparámetros del Cuadro 5.4. En el Cuadro 5.5 se definen los intervalos sobre los cuales aplicaremos grid-search, usando como referencia el método de Smith. Dando como resultado el Cuadro 5.6.

Cuadro 5.5: Grid-Search

Parámetros	Step Decay	Exponential Decay	Time-Based Decay
base_lr	$[10^{-2}, 10^{-4}]$	$[10^{-2}, 10^{-4}]$	$[10^{-2}, 10^{-4}]$
decay	[0,9, 0,95]	[0,007, 0,002]	[0,01, 8]
epoch_drop	5, 10, 20, 40, 50	-	-

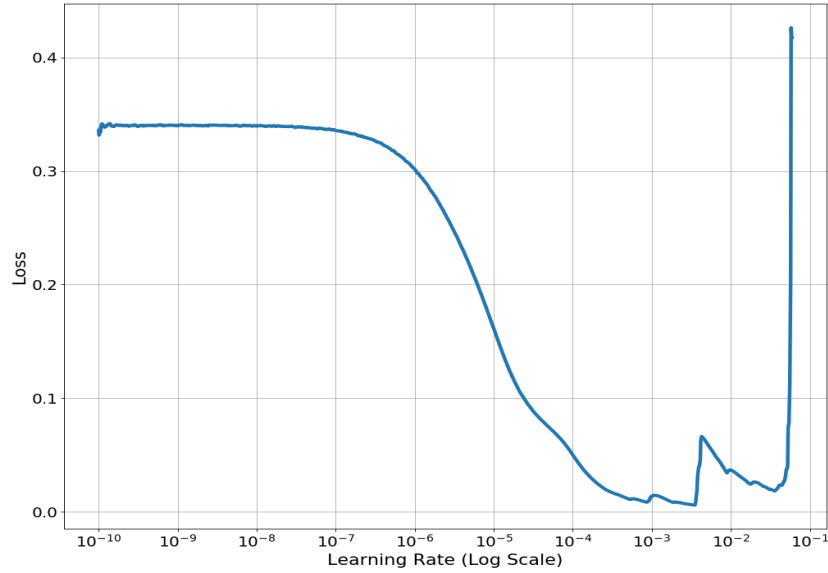


Figura 5.1: Método de Smith

Cuadro 5.6: Parámetros de los Algoritmos de Decrecimiento

Parámetros	Step Decay	Exponential Decay	Time-Based Decay
base_lr	5×10^{-4}	0,0005	0,005
decay	0,9	0,002	0,875
epoch_drop	20	-	-

Observando los parámetros del Cuadro 5.6 se procedió a entrenar la red por 1000 épocas utilizando los algoritmos de decrecimiento de la tasa de aprendizaje y los hiperparámetros del Cuadro 5.4 antes mencionados. Los compararemos mediante su ECM para obtener el mejor algoritmo de decrecimiento. Como se puede observar en la Figura 5.2 Step Decay es el algoritmo que mejor resultado obtuvo.

Luego comparamos Step Decay con Cyclical Decay [18], donde la tasa de aprendizaje varía entre un máximo de 5×10^{-3} y un mínimo de 5×10^{-6} , con un paso cada 8 épocas aproximadamente (ver Figura 2.11). Observando la Figura 5.3, podemos concluir que Step Decay da un mejor resultado.

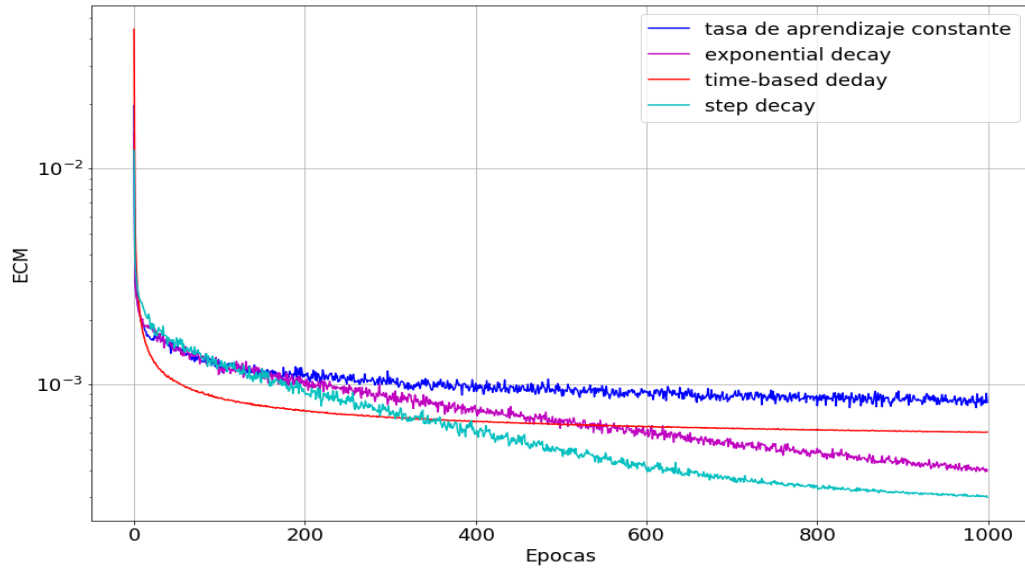


Figura 5.2: Comparación de los algoritmos de decrecimiento de la tasa de aprendizaje

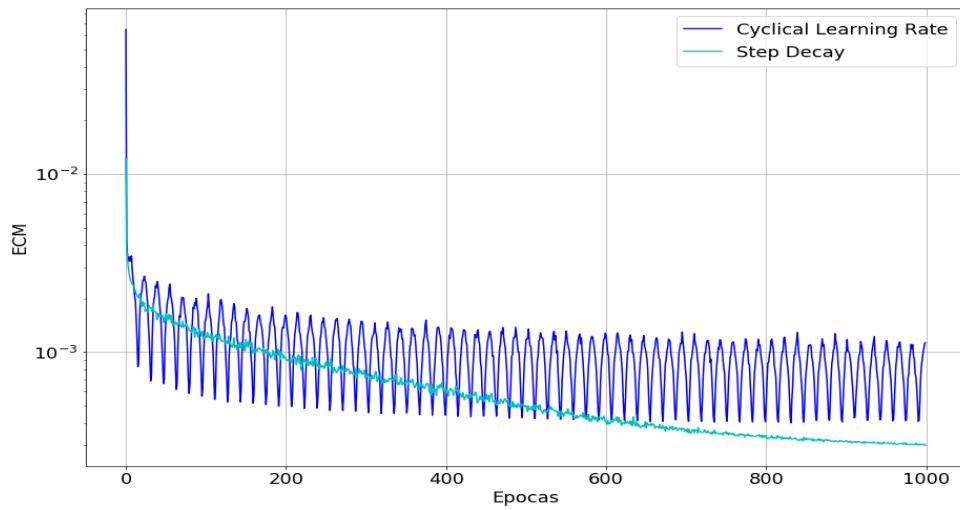


Figura 5.3: Cyclical Learning Rate vs Step Decay

5.4. Optimización

El objetivo aquí es aprender la relación existente entre las volatilidades implícitas y los precios de las opciones. Vemos que la derivada de la fórmula de Black-Scholes 4.3 sobre σ puede volverse arbitrariamente pequeña, por lo tanto su recíproca es muy grande lo que puede dar lugar a un problema de gradiente pronunciado.

Ahora bien, se pueden generar importantes errores de predicción en regiones con grandes gradientes. Por lo tanto, proponemos un enfoque de aplanamiento del gradiente para manejar este problema.

Primero, cada opción puede ser dividida entre el valor intrínseco y un valor temporal. El valor intrínseco de una opción en un tiempo t es el payoff que se obtiene si se ejerciera en ese momento. Para una call sería $\max(S_t - K, 0)$. Aquí es como que considera el K descontado con la tasa r .

Luego sustraemos el valor intrínseco de la siguiente manera:

$$\underbrace{\tilde{c}}_{\text{valor temporal}} = c - \underbrace{\max(S_0 - Ke^{-r\tau}, 0)}_{\text{valor intrínseco}}.$$

En nuestro caso sería:

$$\frac{\tilde{c}}{K} = \frac{c}{K} - \max(S/K - e^{-r\tau}, 0) \quad (5.7)$$

El nuevo cálculo propone superar el problema, logrando reducir el gradiente pronunciado aplicando una transformación logarítmica sobre el valor de la opción, obteniendo como entrada de la red las uplas $\{\ln(c/K), S_0/K, r, \tau\}$ [1].

Capítulo 6

Resultados

En este capítulo vamos a analizar la efectividad de los modelos propuestos. El experimento se realizó en CPU(Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz)

6.1. Red inicial

Con los hiperparámetros obtenidos en el Capítulo 5, se definieron las muestras de entrenamiento (10^6 elementos), validación (10^5 elementos) y test (10^5 elementos). Se crearon dos muestras, una amplia y una estrecha. La muestra amplia es para entrenamiento y validación y la muestra estrecha para test. En el Cuadro 6.1 se define el ambiente de las muestras, y para generar la muestra se utilizó el algoritmo 5.1.

Cuadro 6.1: Hiperparámetros de la Muestra

	Parámetros	muestra amplia	muestra estrecha
Entrada	precio ratio(S_0/K)	[0,4, 1,6]	[0,5, 1,5]
	Tiempo de madurez(τ)	[0,2, 1,1]	[0,3, 0,95]
	volatilidad(σ)	[0,01, 1]	[0,02, 0,9]
	Tasa libre de riesgo(r)	[0,02, 0,1]	[0,03, 0,08]
Salida	Precio de Call(c/K)	(0, 0,9)	(0, 0,73)

Una vez obtenida la muestra, la entrada de la red fueron las uplas $\{c/K, S_0/K, r, \tau\}$ y la salida $\{\sigma\}$. Se entrenó la red durante 1000 épocas utilizando la muestra amplia y los hiperparámetros del obtenidos Capítulo anterior.

Una vez entrenada la red, en el Cuadro 6.2 y en las Figuras 6.1 y 6.2 se pueden observar los resultados una vez realizado el entrenamiento de la red, donde $S(\sigma)$ es la desviación estándar de la diferencia entre la volatilidad implícita y la volatilidad estimada por la red.

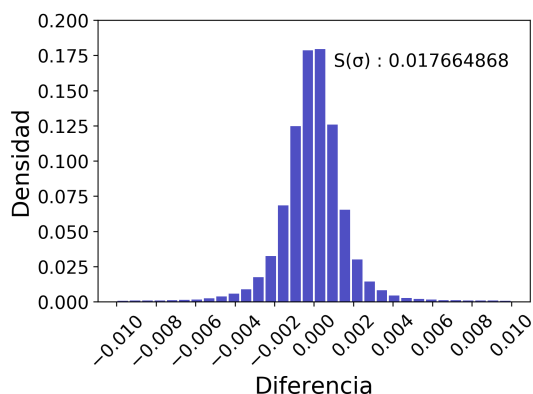


Figura 6.1: Predicción muestra amplia.

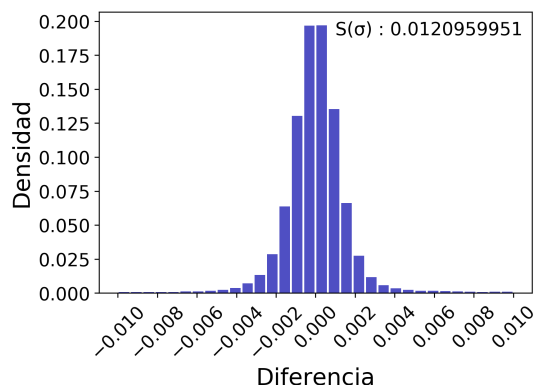


Figura 6.2: Predicción muestra estrecha.

Cuadro 6.2: Error de predicción en el test de la red original.

	ECM	EAM	EPAM	R^2
Muestra Amplia	$3,21 \times 10^{-4}$	$6,09 \times 10^{-3}$	9,30	0,996031
Muestra Estrecha	$1,47 \times 10^{-4}$	$4,07 \times 10^{-3}$	5,46	0,997714

6.2. Optimización

Ya tenemos los hiperparámetros óptimos del capítulo anterior, entonces vamos a definir la muestra de entrenamiento (10^6 elementos), validación (10^5 elementos) y test (10^5 elementos). Crearemos 2 muestras, una amplia y una estrecha. A diferencia de la muestra de la sección anterior, en esta se va a modificar utilizando la fórmula (5.7) del capítulo anterior. En el cuadro 6.3 se define el ambiente de la muestra, y para generar las muestras se usa el algoritmo 5.1.

Cuadro 6.3: Hiperparámetros de la Muestra

	Parámetros	muestra amplia	muestra estrecha
Entrada	precio ratio(S_0/K)	[0,4, 1,6]	[0,5, 1,5]
	Tiempo de madurez(τ)	[0,2, 1,1]	[0,3, 0,95]
	volatilidad(σ)	[0,01, 1]	[0,02, 0,9]
	Tasa libre de riesgo(r)	[0,02, 0,1]	[0,03, 0,08]
Salida	Precio de Call($\ln(\tilde{c}/K)$)	[-16,12, -0,94]	[-16,12, -0,94]

Una vez obtenida la muestra, la entrada de la red será $\{\ln(\tilde{c}/K, S_0/K, r, \tau)\}$ y la salida $\{\sigma\}$. Vamos a entrenar la red durante 1000 épocas.

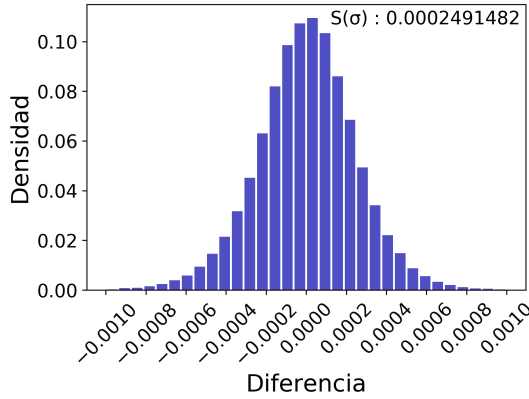


Figura 6.3: Predicción muestra amplia.

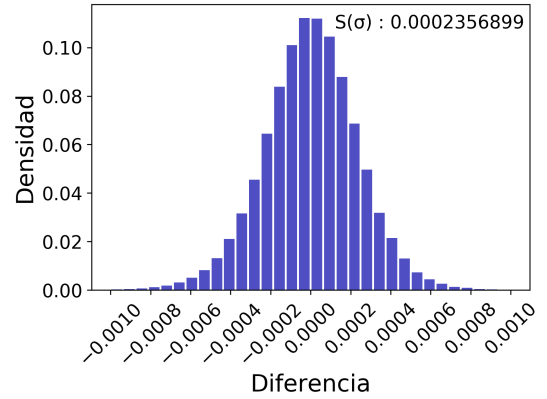


Figura 6.4: Predicción muestra estrecha.

Una vez entrenada la red, en el Cuadro 6.4 y en las Figuras 6.3 y 6.4 se pueden observar los resultados, donde $S(\sigma)$ es la desviación estándar de la diferencia entre la volatilidad implícita y la volatilidad estimada por la red.

Cuadro 6.4: Error de predicción en el test de la red optimizada.

	ECM	EAM	EPAM	R^2
Muestra Amplia	$6,24 \times 10^{-8}$	$1,92 \times 10^{-4}$	0,059	0,999999104
Muestra Estrecha	$5,58 \times 10^{-8}$	$1,84 \times 10^{-4}$	0,062	0,999999020

6.2.1. Cambio en función de decrecimiento de la Tasa de Aprendizaje

Observando la Figura 5.1 podemos concluir que el entrenamiento de la red con una tasa de aprendizaje menor a 10^{-7} prácticamente no actualizaría los pesos, entonces si queremos hacer mas épocas en el entrenamiento de la red, con el algoritmo que usamos en la sección anterior tendríamos en la época 3000 una tasa de aprendizaje aproximado de $1,87 \times 10^{-17}$, y en la época 1000 la tasa de aprendizaje aproximado sería de $2,65 \times 10^{-8}$. Una posible solución a este problema es modificar los hiperparámetros óptimos aumentando el *base_lr*, aumentar el *decay*, y aumentar el *epoch_drop*, para que la tasa de aprendizaje descienda mas lentamente. Otra solución sería cuando la tasa de aprendizaje sea menor a un umbral, aumentar el *base_lr*. El umbral puede ser variable.

El algoritmo 6.1 muestra el método de decrecimiento de la tasa de aprendizaje que

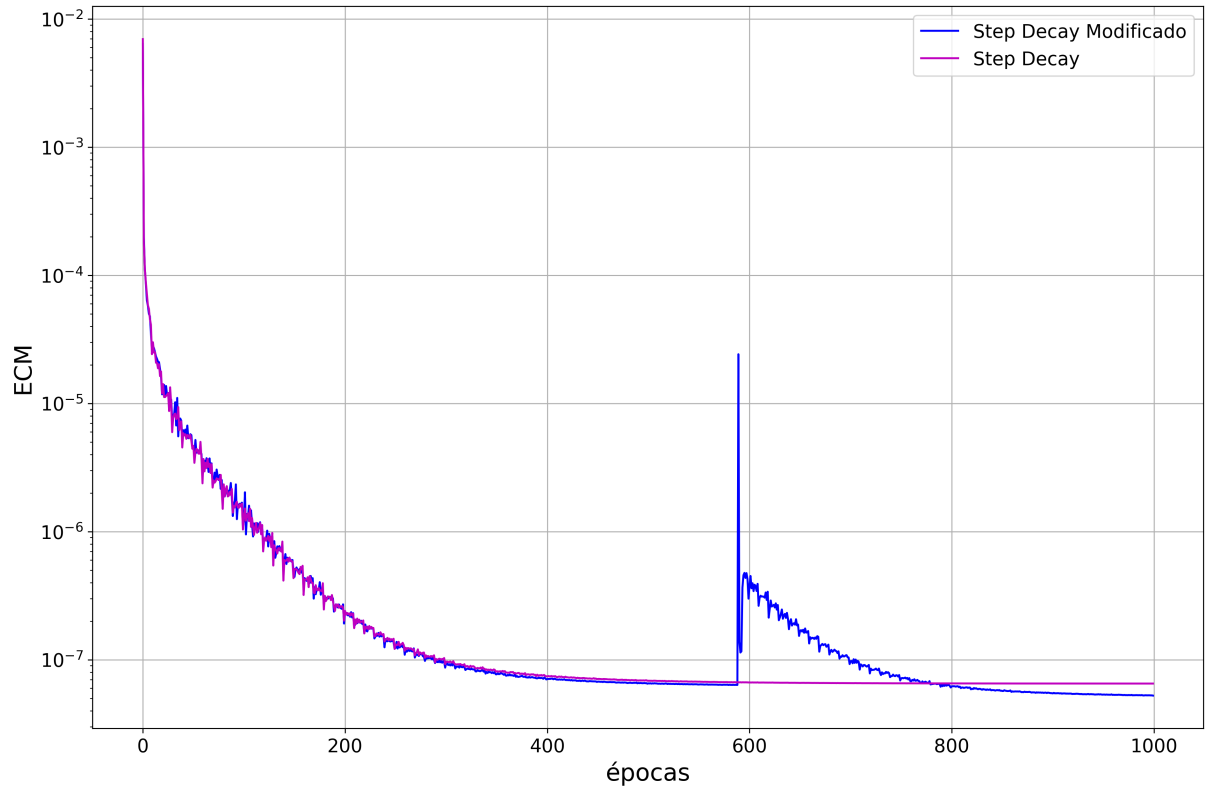


Figura 6.5: Step Decay contra Step Decay Modificado

utilizaremos.

Algorithm 6.1: Step Decay Modificado

Input : epoch
Output: lrate

```

1  $i := 0$ ;
2  $step := \text{floor}((1+epoch)/10)$ ;
3  $lrate := 0.0005 * \text{pow}(0.9, step)$ ;
4 while  $lrate < 10^{-(6+i)}$  do
5   |  $lrate := 100 * lrate$ ;
6   |  $i = i + 0.5$ ;
7 end
8 return;

```

La Figura 6.5 muestra la comparación entre el step decay usado anteriormente y el step decay modificado.

Las Figuras 6.6 y 6.7 muestran el tasa de aprendizaje resultante con los diferentes

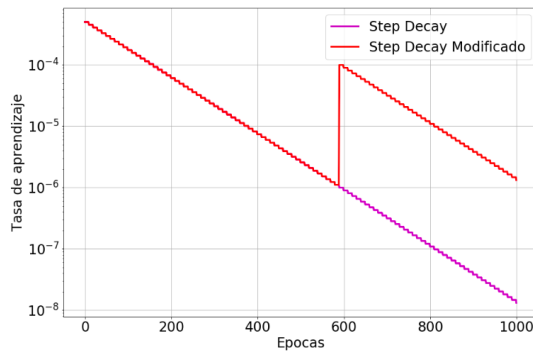


Figura 6.6: Tasa de aprendizaje sobre 1000 épocas.

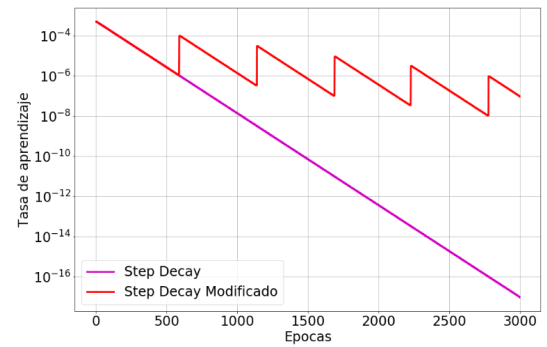


Figura 6.7: Tasa de aprendizaje sobre 3000 épocas.

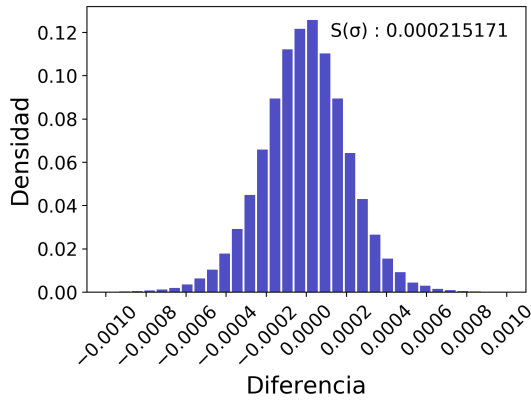


Figura 6.8: Predicción muestra amplia.

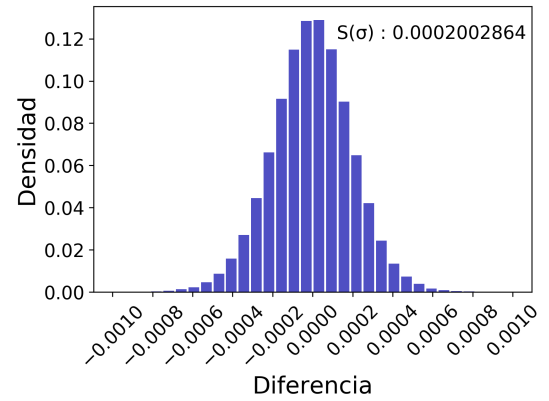


Figura 6.9: Predicción muestra estrecha.

algoritmos.

Luego vamos a entrenar la red durante 3000 épocas usando step decay modificado. Una vez entrenada la red en el Cuadro 6.5 y en las Figuras 6.8, 6.9 se observan los resultados, donde $S(\sigma)$ es la desviación estándar de la diferencia entre la volatilidad implícita y la volatilidad estimada por la red.

Cuadro 6.5: Error de predicción en el test de la red final.

	ECM	EAM	EPAM	R^2
Muestra Amplia	$4,67 \times 10^{-8}$	$1,66 \times 10^{-4}$	0,0515	0,999999329
Muestra Estrecha	$4,02 \times 10^{-8}$	$1,57 \times 10^{-4}$	0,0526	0,999999295

En el Cuadro 6.6 se compara el rendimiento de cada red neuronal implementada por su error y por su desviación estándar, se puede observar que cuando se modifica la entrada de la red en la sección 6.2 las métricas de la red mejoran considerablemente (4 ordenes en el ECM), y cuando se modifica la función de decrecimiento de la tasa de aprendizaje mejora aproximadamente un 25 % comparando su ECM.

Cuadro 6.6: Error de predicción en el test de la red final.

	ECM	EAM	EPAM	R^2	Desvió estándar
Red inicial	$3,21 \times 10^{-4}$	$6,09 \times 10^{-3}$	9,30	0,996031	0,017664868
Red optimizada	$6,24 \times 10^{-8}$	$1,92 \times 10^{-4}$	0,059	0,999999104	0,0002491482
Red definitiva	$4,67 \times 10^{-8}$	$1,66 \times 10^{-4}$	0,0515	0,999999329	0,000215171

6.3. Métodos Numéricos

6.3.1. Problemas Numéricos

En la muestra amplia definida en la sección 6.1 hay aproximadamente un 0.05 % de casos que no cumplen con la condición $g(a)g(b) < 0$, para $a \rightarrow 0$ y $b \rightarrow \infty$, donde g es la función 4.1 anteriormente.

El problema radica en el cálculo de la fórmula de Black-Scholes definida en la ecuación 2.1, básicamente en la distribución de probabilidad acumulada $\Phi(d)$. Por definición $\Phi(d) < 1$, $\forall d \in \mathbb{R}$. Pero Python usa aritmética de punto flotante IEEE-754 [22], donde su precisión es 2^{-56} , dando así una precisión aproximada de 15,95 dígitos decimales. Luego usando la función de Scipy para el cálculo de la distribución normal acumulada definida en la referencia [21], obtenemos $\Phi(n) = 1$, para $n > 8,3$. Volviendo a la fórmula de Black-Scholes, si bien sabemos que $\Phi(d_1) > \Phi(d_2)$, ya que $\sigma, T > 0$, pero hay casos en que el extremo inferior definido para aplicar el método de bisección (o Brent), se obtiene $\Phi(d_1) = \Phi(d_2)$ por el problema de precisión, ocasionando el error antes mencionado.

Luego el problema está en la cantidad de posibles números representables en un dado rango. El formato se escribe con un significando que tiene un bit entero implícito de valor 1 (excepto para los números especiales). Con los 52 bits de la mantisa, la precisión total es por lo tanto de 53 bits (es decir de $53 \log_{10}(2) \approx 15,955$ que se redondea a 16 dígitos decimales). El exponente de este formato está sesgado o desplazado en 1023 unidades, ya que como el máximo valor representado por 11 bits es $2^{11} - 1 = 2047$, es la mitad de este rango la que representa exponentes positivos y la otra, exponentes negativos [22]. Podemos observar esto en la Figura 6.10. Lo expuesto permite ver que hay 2^{52} números de punto flotante entre 2^n y 2^{n+1} para todo n entero en $[-1023, 1022]$.

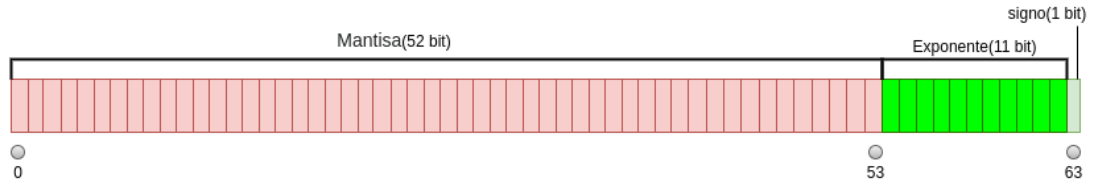


Figura 6.10: Estructura de un número en formato de coma flotante de doble precisión

Por ejemplo, consideremos un caso en particular, tomando los valores del Cuadro 6.7 se puede observar en la Figura 6.11 que $g(n) \geq 0$ para $n \in [0,01, 1]$, por lo tanto no se puede aplicar el método de bisección (o Brent).

c :	5,983489610184446
S :	15,752756180327959
k :	10
r :	0,09010364215460305
T :	0,2590760904347537

Cuadro 6.7: Parámetros que no cumplen la pre-condición de los métodos numéricos

Luego si $\sigma = 0,01$ (extremo inferior del intervalo), se obtiene, $\Phi(d_1) = 1$, $\Phi(d_2) = 1$ y $g(0,01) = 1,7763568394002505 \times 10^{-15}$. Pero si se evalúa la función de distribución normal acumulada utilizando la volatilidad implícita ($\sigma = 0,11928197090875538$), se obtiene $\Phi(d_1) = 0,9999999999999986$ (Figuras 6.12, 6.13), $\Phi(d_2) = 0,9999999999999982$ y $g(0,11928197091) = 0$, se contradice la propiedad de monotonía creciente.

6.3.2. Precisión Métodos Numéricos

En la sección 6.3.1 vimos que hay casos en los que no se puede aplicar los métodos numéricos, exceptuando esos casos, vamos a analizar su precisión. Además se vio que lo máximo que podemos aspirar es un error absoluto de 2^{-56} . Ahora vamos a analizar si en la práctica, la tolerancia de los métodos corresponden con los resultados obtenidos de aplicar los mismos. En el método de Brent utilizamos el de la biblioteca Scipy [29].

Cuadro 6.8: Error Brent y Bisección

	ECM	EAM	EPAM	R^2
Bisección	$1,10 \times 10^{-8}$	$2,89 \times 10^{-6}$	0,00469	0,99999986
Brent	$8,17 \times 10^{-6}$	$9,18 \times 10^{-5}$	0,35894	0,99989551

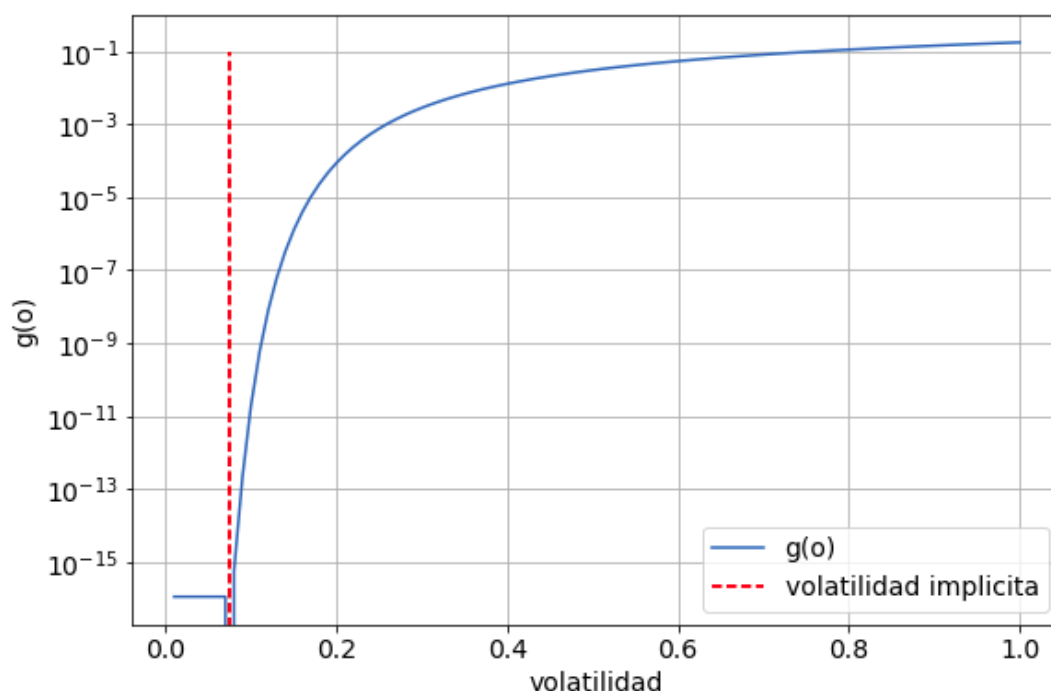
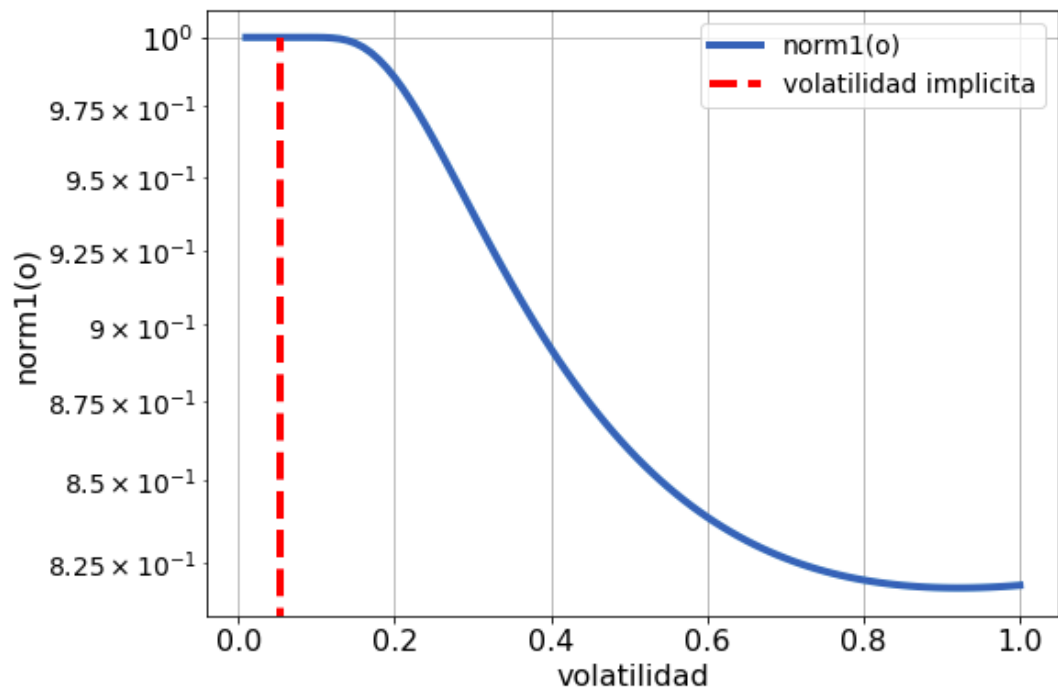
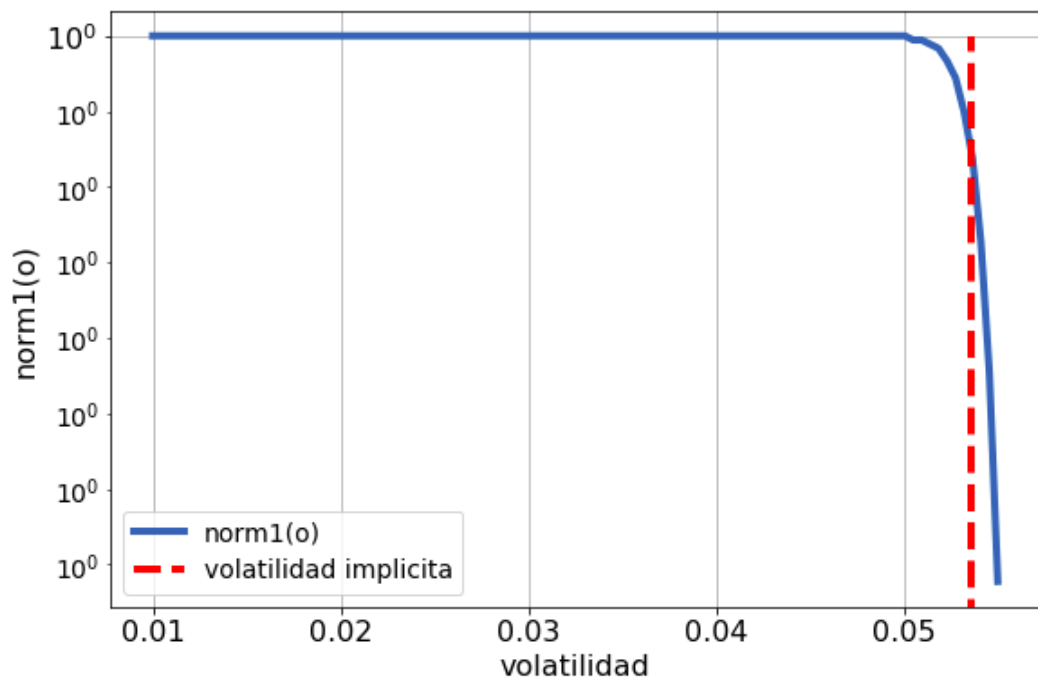


Figura 6.11: Función g .

Observando el Cuadro 6.8, el error absoluto medio del método de bisección es $2,89 \times 10^{-6}$ y de Brent es $9,18 \times 10^{-5}$, siendo la tolerancia 2^{-56} , esto es por el problema de precisión y el impacto que tiene la volatilidad sobre el precio de la opción (derivada de la fórmula de Black-Scholes sobre σ) utilizando la fórmula de Black-Scholes 2.1. En la Figura 6.14, que compara el precio de la opción con respecto a la variación de volatilidad, utilizando diferentes ratios (S/K), se puede observar que mientras más diferencia hay entre S_0 y K la volatilidad tiene menos impacto sobre el precio en especial en los casos OTM y en las volatilidades cercanas a 0. Luego en la Figura 6.15, que compara el precio de la opción con respecto a la variación de volatilidad, utilizando diferentes tasas de interés, se observa prácticamente un desplazamiento. Por último en la Figura 6.16, que compara el precio de la opción con respecto a la variación de volatilidad, utilizando diferentes tiempos de madurez, se puede observar que mientras los contratos sean más largos la volatilidad tiene un mayor impacto sobre el precio.

Ahora bien observando los gráficos podemos concluir que la volatilidad tiene menos impacto en los contratos cortos(en nuestro ambiente), en OTM y con volatilidades menores a 0.2, pero en la practica los casos con mas error son ITM con $\Phi_1(\sigma)$ y $\Phi_2(\sigma)$ cercanos a 1.

Esto se debe a la densidad de puntos flotantes visto en la sección anterior. Luego como el ancho de 11 bits del exponente permite la representación de números en el rango

Figura 6.12: $\Phi(d_1)$.Figura 6.13: $\Phi(d_1)$.

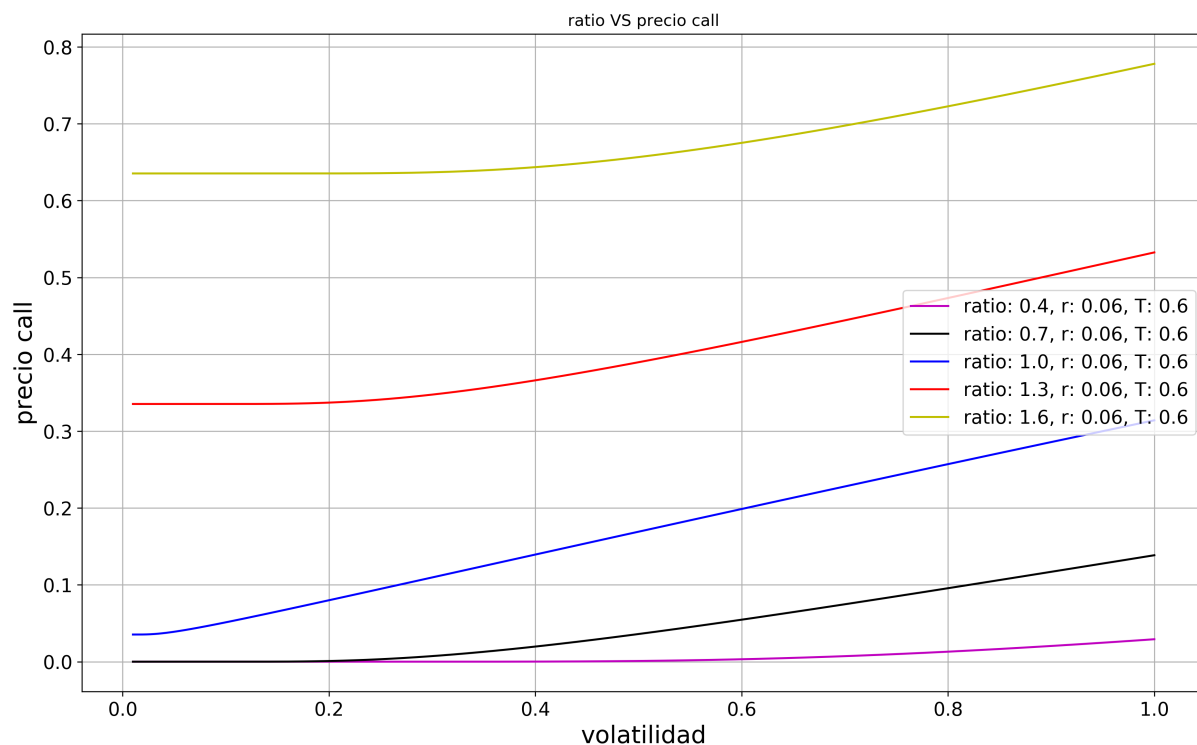


Figura 6.14: Volatilidad implícita contra precio call con distintos ratios

comprendido entre 2^{-1023} y 2^{1023} (10^{-308} y 10^{308} aproximado). Esto nos permite concluir que hay aproximadamente la misma cantidad de números representables entre 10^{-307} y 10^{-306} que en 0,1 y 1, osea que su densidad es mayor cuando tiende a 0 (Observar Figura 6.17). Por lo tanto los $\Phi_1(\sigma)$ y $\Phi_2(\sigma)$ cercanos a 0 en OTM van a tener mas precisión aunque el peso de la volatilidad sobre el precio de la opción sea menor.

6.4. Tiempo de Ejecución y Robustez

Una vez entrenada la red (Fue calculado sobre GPU(GeForce GTX 1080/PCIe/SSE2), RAM(15.6 GiB). La búsqueda de hiperpámetros en un tiempo 10,08 horas, la búsqueda del algoritmo de la tasa de aprendizaje fue 11,72 horas y entrenamiento de la red 3000 épocas fue 8,83 horas), vamos a observar el tiempo ejecución de evaluación de la misma sobre una muestra de tamaño 10^5 . Además el tiempo de ejecución de los métodos de Brent y bisección sobre la misma muestra en una CPU(Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz).

También analizaremos su robustez, esto sería, si para todo elemento de la muestra el

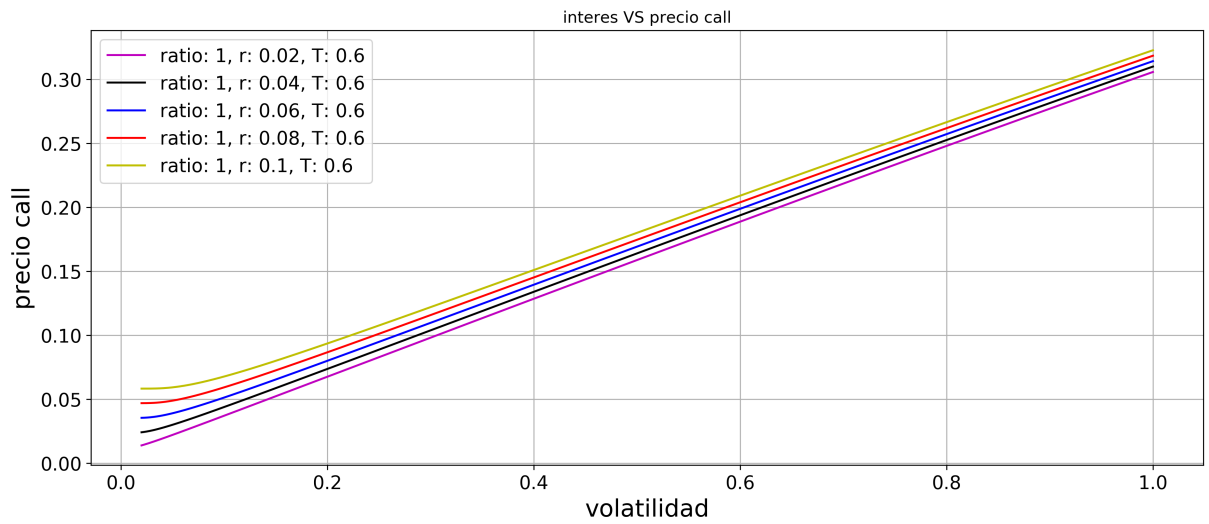


Figura 6.15: Volatilidad implícita contra precio call con distintas tasas de interés

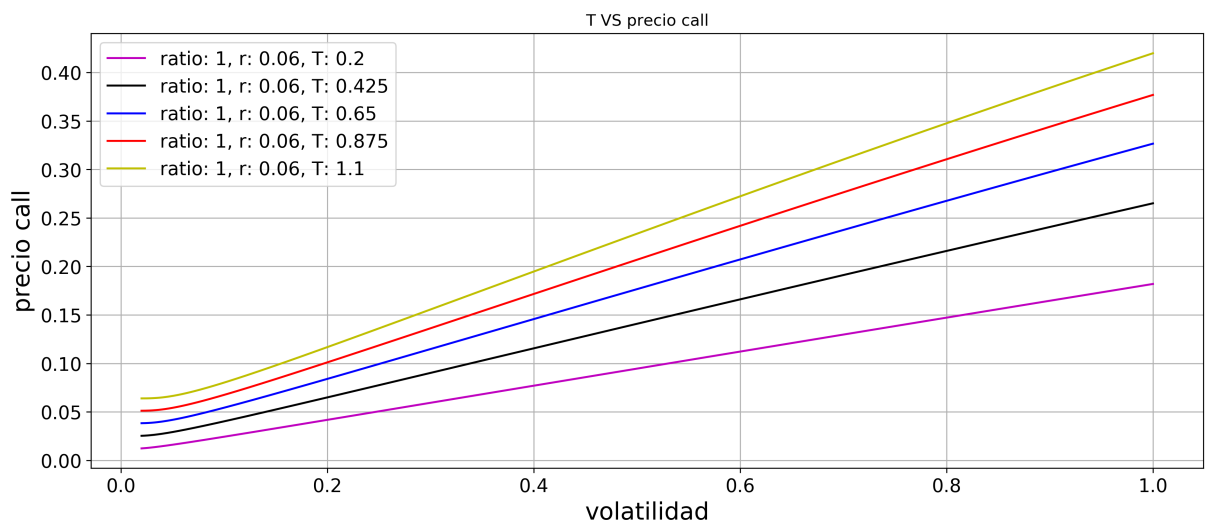


Figura 6.16: Volatilidad implícita contra precio call con distintos tiempos de madurez

modelo genera una salida.

En el Cuadro 6.9 se observan el tiempo de ejecución y la Robustez.

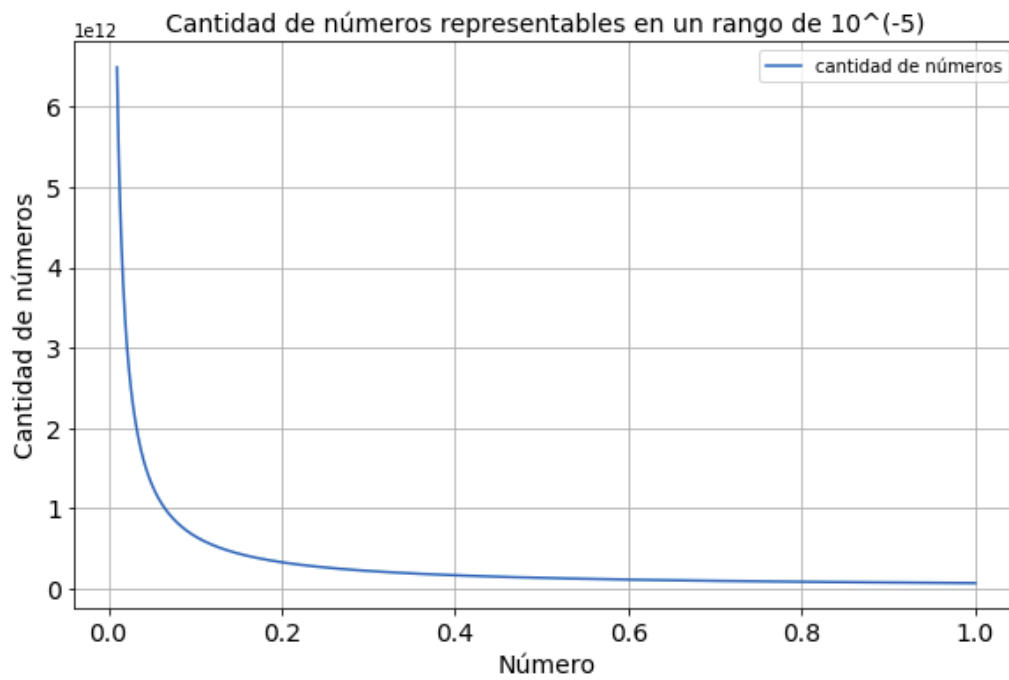


Figura 6.17: Densidad de números representables en rango de 10^{-5} entre 0.1 y 1

Cuadro 6.9: Robustez y Tiempo de Ejecución

	Red Neuronal	Método de Brent	Método de Bisección
Tiempo de Ejecución(s)	7,34	157,05	318,31
Robustez	Si	No	No

Capítulo 7

Conclusiones

En este trabajo se presentó la importancia del cálculo de la volatilidad implícita. Además se propuso una manera diferente del cálculo que usualmente se utiliza, usando redes neuronales feedforward.

Si bien el método de bisección estima mejor la volatilidad implícita, no es una mejora significativa con respecto a la red neuronal o al método de Brent, sin embargo en tiempo de ejecución la red neuronal es 21 veces menor que el método de Brent y 43 veces menor que el método de bisección.

En caso de tener datos atípicos, por ejemplo precio de strike tres veces mas grande que el precio del subyacente, o tasas libre de riesgo superiores al 20 %, o contratos muy largos, en esos casos es conveniente usar los métodos numéricos, ya que la red no va a estimar bien, porque la red no entrenó con esos datos. Sin embargo hay casos en los que los métodos numéricos no pueden ser aplicados.

En caso que la precisión sea imprescindible, entonces se utilizará el método de bisección.

Dentro de las posibilidades de continuar con el trabajo tenemos, uso de GPU en la implementación de los modelos numéricos propuestos, presentar soluciones para el problema numérico del capítulo 6.

En este trabajo se planteó el cálculo de la volatilidad implícita mediante opciones call europeas, además se podrían proponer otros modelos obtener la volatilidad implícita mediante puts europeas.

Bibliografía

- [1] Shuaiqiang Liu, Cornelis Oosterlee, and Sander Bohte. Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):16, Feb 2019.
- [2] Ramazan Gençay, René Garcia. Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics*, 94:93–115, Enero 2000.
- [3] James M. Hutchinson, Andrew W. L, Tomaso Poggio. A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *Journal of Finance*, 49:851–889, Julio 1994.
- [4] Patricia Kisbye. Modelos matemáticos en finanzas cuantitativas. FAMA-UNC, 2019.
- [5] Mark Yallop. The future of the otc markets. https://thehill.com/sites/default/files/ICAP_TheFutureoftheOTCMarkets_0.pdf, 2008.
- [6] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *The Journal of Political Economy*, 81(3):637–654, mayo 1973.
- [7] John C. Hull. *Options, Futures, and Other Derivatives*, páginas 381–387. Pearson Prentice Hall, séptima edición, 2009.
- [8] Ibíd., páginas 387–389.
- [9] Ward Cheney y David Kincaid. *Métodos numéricos y computación*, páginas 78. Cengage Learning, sexta edición, 2011.
- [10] Ibíd., páginas 76–82.
- [11] Ward Cheney y David Kincaid, op. cit., páginas 111–112.
- [12] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes The Art of Scientific Computing*. páginas 454–456. Cambrindge University Press, tercera edición, 2007
- [13] Tom M. Mitchell. *Machine Learning*. páginas 86–88. McGraw-Hill Education, 1997

- [14] John A. Hertz, Anders S. Krogh, Richard G. Palmer. *Introduction To The Theory Of Neural Computation*. páginas 90–92. Westview Press, 1991.
- [15] Wei Di, Anurag Bhardwaj, Jianing Wei. *Deep Learning Essentials*. páginas 61–62. Packt Publishing, 2018.
- [16] Jason Brownlee. *Deep Learning With Python*. páginas 109–112. (2016),
- [17] Wei Di, Anurag Bhardwaj, Jianing Wei, op. cit., páginas 219.
- [18] Leslie N. Smith. Cyclical Learning Rates for Training Neural Networks. *ArXiv:abs/1506.01186*. <https://arxiv.org/abs/1506.01186>, 2015.
- [19] Prosper Lamothe Fernández, Miguel Pérez Somalo. *Opciones Financieras y Productos Estructurados*. páginas 319–336. McGraw-Hill Interamericana de España S.L., 2003.
- [20] Jay Kaeppel. *The Option Trader's Guide to Probability, Volatility, And Timing*. páginas 156–201. John Wiley & Sons, Inc, 2002
- [21] Distribución Normal Acumulada. <https://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.stats.norm.html>.
- [22] Densidad Punto Flotante. <https://web.archive.org/web/20160806053349/http://www.csee.umbc.edu/~tsimo1/CMSC455/IEEE-754-2008.pdf>.
- [23] Función de Activación. <https://keras.io/activations/>
- [24] Inicialización de Pesos. <https://keras.io/initializers/>
- [25] Optimizadores del Descenso por el Gradiente. <https://keras.io/optimizers/>
- [26] Función de Error. <https://keras.io/losses/>
- [27] Modelo Secuencial. <https://keras.io/api/models/sequential/>
- [28] Grid Search. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV
- [29] Método de Brent. <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.brentq.html>