

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PUCP

**Implementación de un algoritmo de aprendizaje profundo basado en
eventos para el problema de predicción de movimiento bursátil
Tesis para obtener el título profesional de Ingeniero Informático**

AUTOR:

Jaime Diego Bustamante Arce

ASESOR:

César Armando Beltrán Casteñon

Lima, Junio, 2021

RESUMEN DE TESIS

La predicción de precios bursátiles, acciones e índices siempre ha sido un tema de interés en el mundo financiero, no solo por su capacidad de originar grandes rentabilidades en poco tiempo, sino también por su volatilidad y complejidad. Así, desde que los mercados bursátiles fueron concebidos diferentes investigadores en variadas áreas han tratado de “vencerlo” prediciendo su comportamiento, como el índice S&P 500 que lista la cotización de las 500 corporaciones más líquidas de la Bolsa de New York. Uno de los enfoques es el fundamentalista, que busca predecirlo de acuerdo a las noticias en los medios de las empresas listadas en la Bolsa de Valores. Desde el lado informático, diversas técnicas han venido siendo aplicadas para realizar esta predicción como estadísticas y las clásicas herramientas de aprendizaje de máquina. Sin embargo, con el creciente aumento de volumen de información, se hace necesario aplicar técnicas que consigan lidiar con esta información no estructurada. Técnicas como redes profundas recurrentes (LSTM), se han mostrado ad-hoc para el manejo de información temporal, debido a que tienen de capacidad de memorizar hechos pasados, que persisten en el tiempo. En el presente trabajo se propone una metodología y conjunto de redes neuronales profundas para la predicción de movimiento bursátil a partir de eventos y noticias corporativas. Para ello no solo se considera la contextualización de palabras, sino también sus relaciones y composición semántica, estructura e historia para la predicción del índice S&P 500.

En resumen, el presente proyecto obtiene resultados exitosos puesto que sobrepasan a los del estado del arte. Así, el conjunto de modelos neuronales propuestos puede ser usados como apoyo en la decisión de inversión diaria en el índice S&P 500.

Contenido

Índice de ecuaciones	5
Índice de imágenes.....	6
Índice de tablas	7
Índice de Pseudocódigos.....	8
1. Problemática	9
2. Marco Teórico	12
2.1. Conceptos generales de procesamiento natural de lenguaje	12
2.2. Conceptos generales de aprendizaje de máquina	13
2.3. Aprendizaje supervisado	13
2.3.1. Máquinas de vectores de soporte.....	14
2.3.2. Redes neuronales artificiales	14
2.3.3. Modelos de aprendizaje profundo	14
2.3.3.1. Redes recurrentes profundas y Memoria de largo / corto plazo (LSTM).....	15
2.3.3.2. Redes neuronales tensoriales	16
2.3.4. Indicadores de desempeño.....	16
2.3.4.1. Coeficiente de correlación de Matthews.....	16
2.3.4.2. Rentabilidad y estrategias de inversión.....	16
2.3.5. GPU	17
2.4. Conceptos generales en la teoría de la predicción de valores	17
2.4.1. Hipótesis del caminante aleatorio.....	17
2.4.2. Hipótesis del mercado informacionalmente eficiente	17
2.5. Conceptos generales mercado de valores.....	18
2.5.1. Acción.....	18
2.5.2. Mercado de valores.....	18
2.5.3. Volatilidad	18
2.5.4. Liquidez.....	18
2.5.5. Cotización.....	19
2.6. Conclusión.....	19
3. Estado del arte.	20
3.1. Métodos manuales.....	20
3.2. Proyectos de investigación para la predicción de valores futuros.....	20
3.3. Productos comerciales.....	23
3.4. Conclusiones	23
4. Objetivo general	25
5. Objetivos específicos	26
6. Resultados esperados.....	27
7. Herramientas y metodología.....	28

7.1.	Herramientas	29
7.1.1.	IntelliJ	29
7.1.2.	Python.....	29
7.1.3.	Deeplearning4J	29
7.1.4.	GPU Nvidia Grid – Arquitectura Kepler	30
7.1.5.	Standford Extractor de información abierto (Open Information Extractor).....	30
7.1.6.	Word2Vec.....	30
7.1.7.	Microsoft Excel	30
7.1.8.	Java	31
7.1.9.	Microsoft Visio.....	31
7.1.10.	OpenCSV	31
7.1.11.	Scrapy	31
7.2.	Métodos.....	31
7.2.1.	Precisión de pronóstico.....	32
7.2.2.	Coeficiente de correlación Matthews	32
7.2.3.	Estrategia de inversión.....	32
7.2.4.	Análisis de ganancia neta anualizada	33
7.3.	Metodología y plan de trabajo.....	33
7.3.1.	Estudio de la literatura	33
7.3.2.	Extracción y estructuración de la información	33
7.3.3.	Representación vectorial de los eventos	33
7.3.4.	Diseño e implementación de la red neuronal propuesta	34
7.3.5.	Entrenamiento.....	34
7.3.6.	Evaluación de la red neuronal propuesta	34
7.3.7.	Conclusiones.....	35
8.	Alcance, limitaciones y riesgos	36
8.1.	Limitaciones	36
8.2.	Riesgos	37
9.	Justificación y viabilidad.....	38
9.1.	Justificación del proyecto de fin de carrera.....	38
9.2.	Análisis de la viabilidad del proyecto de fin de carrera	38
9.2.1.	Viabilidad técnica	39
9.2.2.	Viabilidad temporal	40
9.2.3.	Viabilidad económica	40
9.2.4.	Conclusión.....	41
10.	Extracción y estructuración de información	43
10.1.	Descripción.....	43
10.1.1.	Recolección de información	43

10.1.2.	Extracción de eventos	44
10.1.3.	Obtención de parámetros adicionales	45
11.	Representación vectorial de eventos	46
11.1.	Descripción	46
11.2.	Representación vectorial de palabras	46
11.3.	Representación vectorial de relaciones.....	48
12.	Modelo de red neuronal recurrente profunda con memoria Largo-Corto plazo	51
12.1.	Arquitectura de red neuronal	51
12.2.	Implementación e integración del módulo	54
12.3.	Entrenamiento y validación cruzada con K iteraciones.....	56
12.4.	Implementación de la simulación en el mercado bursátil.....	57
13.	Experimentación y resultados	62
13.1.	Observaciones y configuración de la experimentación	62
13.2.	Experimentación sobre modelos predictivos.....	63
13.3.	Experimentación del modelo global	65
13.3.1.	Experimentación de la predicción en el índice S&P 500.....	65
13.3.2.	Experimentación mediante la simulación de mercado.....	67
14.	Discusión.....	69
14.1.	Sobre la estructuración y estructuración de información	69
14.2.	Sobre la representación vectorial de palabras y eventos	69
14.3.	Sobre el modelo de red recurrente profundo con memoria de Largo-Corto plazo 70	
15.	Conclusiones y trabajos futuros	72
15.1.	Conclusiones.....	72
15.2.	Trabajos futuros y recomendaciones	72
16.	Bibliografía	74

Índice de ecuaciones

Ecuación 1: Coeficiente de correlación de Matthews.....	16
Ecuación 2: Función de evaluación de la red tensorial.....	50
Ecuación 3: Compuerta de olvido [COLAH, 2015]	53
Ecuación 4: Compuerta de estado de célula [COLAH, 2015].....	53
Ecuación 5: Compuerta de salida.....	53
Ecuación 6: Función de costo “Suma de errores cuadráticos” [GUO, 2013]	56
Ecuación 7: Gradiente descendiente para actualizar los pesos de la red neuronal [GUO, 2013]	56
Ecuación 8: Programación del error hacia atrás en el tiempo [GUO, 2013]	57



Índice de imágenes

Imagen 1: Flujo global	35
Imagen 2: Diagrama de Gantt	40
Imagen 3: Flujo de la extracción y estructuración de información.....	43
Imagen 4: Composición semántica [GABOR, 2015]	44
Imagen 5: Representación de palabras [Word2vec, 2017]	46
Imagen 6: Skip-gram [McCorminck, 2016].....	47
Imagen 7: Red neuronal tensorial [SOCHER, 2013].....	49
Imagen 8: Red neuronal tensorial adaptada [DING, 2015]	49
Imagen 9: Arquitectura de red neuronal recurrente	51
Imagen 10: Unidad LSTM, X como entrada y h como salida [COLAH, 2015].....	52
Imagen 11: Efecto de las noticias en la bolsa bursátil [GOOGLE, 2017]	54
Imagen 12: Proceso de “shorting” – Precio decrece [GOOGLE FINANCE, 2017]	58
Imagen 13: Proceso cuando el precio incrementa [GOOGLE FINANCE, 2017]	59
Imagen 14: Red convolucional	64
Imagen 15: Precisión y MCC.....	66



Índice de tablas

Tabla 1: Resultados esperados	28
Tabla 2: Gestión de riesgos.....	37
Tabla 3: Viabilidad económica	41
Tabla 4: Configuración de la arquitectura	52
Tabla 5: Descripción de componentes	54
Tabla 6: Parámetros de los conjuntos de datos	63
Tabla 7: Configuración de la red convolucional.....	64
Tabla 8: Resultados MCC y Precisión sobre los modelos convolucional y recurrente	65
Tabla 9: Parámetros de los conjuntos de datos	66
Tabla 10: Resultados de la experimentación en las medidas de MCC y precisión.....	66
Tabla 11: Rentabilidad por año.....	68
Tabla 12: Comparación versus la literatura actual.....	72



Índice de Pseudocódigos

Pseudocódigo 1: Araña web.....	44
Pseudocódigo 2: Entrenamiento de la red tensorial propuesta [DING, 2014].....	50
Pseudocódigo 3: Implementación de la red neuronal recurrente LSTM	55
Pseudocódigo 4: Implementación de la estrategia de inversión	60



1. Problemática

Una acción es un título emitido por una entidad como parte de su capital social con el objetivo de conferir derechos económicos o políticos [HAMILTON, 2006]. Esta inversión es de renta variable, es decir, que su rentabilidad está intrínsecamente ligada al buen desempeño, así como a los acontecimientos y eventos relacionados con la entidad [HAMILTON, 2006]. Estas acciones son comercializadas en los mercados financieros llamados mercados de valores. Estos mercados están caracterizados fundamentalmente por la falta de plazos de inversión (períodos temporales donde se estudia una inversión y retorno), alta volatilidad (frecuencia de cambios del precio de una acción) y mucha liquidez (cualidad de las acciones de convertirse en efectivo rápidamente) [DOWNES, 2010]. Actualmente, el tamaño de capital asociado a este mercado está estimado en 69 trillones de dólares al cierre del 2015 [CNMV, 2015 – MARKETWATCH, 2015]. Las formas de obtener rentabilidad en las inversiones en capitales son la reevaluación del precio de la acción o pago de dividendos, y mediante el retorno total al accionista [DIGRANDE, 2014].

El primer problema en este contexto es la complejidad para la determinación de valores futuros de las acciones de una entidad, así, una predicción precisa y rápida puede implicar una alta rentabilidad [HAMILTON, 2006]. Con respecto a lo mencionado, la hipótesis del mercado eficiente sugiere que los precios de las acciones reflejan toda la información disponible y pública, y que cualquier cambio en estos precios es consecuencia inmediata de nueva información divulgada [FAMA, 1965]. Sin embargo, debido a la gran cantidad de información financiera no estructurada y no consolidada, la actividad de predicción implica un esfuerzo complejo y, por su naturaleza, ha sido considerada una actividad intensiva humana. De este modo, las metodologías para la predicción de valores caen en tres categorías bien definidas: análisis fundamental, análisis técnico y análisis tecnológico [BORDER, 2014]. Primero, el análisis fundamental busca información (por ejemplo, resultados financieros, noticias, cuentas, etc.) acerca del entorno externo (aspectos macroeconómicos) e interno (desempeño de la empresa estudiada) para estimar el valor futuro que tendrán las acciones [BORDER, 2014]. Segundo, el análisis técnico refiere a la identificación de tendencias de los datos históricos sobre los precios estudiados a través de la búsqueda de patrones del pasado que puedan estar ocurriendo actualmente, para tomar ventaja de estos sin importar las causas que originaron estos comportamientos [BORDER, 2014]. Por último, el análisis tecnológico se preocupa por la implementación de diversos algoritmos para el pronóstico de las tendencias y valores futuros de manera rápida y precisa [BORDER, 2014]. En esta última categoría, los algoritmos que han obtenido mejores resultados están ligados a los algoritmos heurísticos, metaheurísticos y, dentro de estos, los algoritmos que caen en la categoría de “aprendizaje automático” [THAWORNWONG, 2004].

El segundo problema para la inversión en el mercado de valores, tan importante como el primero, radica en la dificultad de un procesamiento rápido y preciso de la información disponible actual para la predicción de precios de las acciones debido a la volatilidad que tiene el mercado de valores [GALLEGO, 2015]. La información obtenida debe ser procesada de la manera más rápida posible para evitar un impacto por costo de oportunidad [GALLEGO, 2015]. Entonces, a consecuencia de la potencia computacional que existe, la literatura y trabajos previos han ido inclinándose por el uso de la metodología tecnológica para abordar la predicción de valores de forma precisa y ágil. De este modo, algoritmos del tipo heurístico / metaheurístico se han preocupado por la agilidad para procesar información, pero con ello pierden la precisión de la predicción. Sin embargo, en los últimos años, con la llegada de las GPU y una reducción del costo de almacenamiento de datos, se ha obtenido más eficiencia en la ejecución de algoritmos más precisos [TENSORFLOW, 2016]. Así, existen algunas herramientas tecnológicas disponibles, como, por ejemplo, PredictWallStreet y StockForecasting, que utilizan métodos de minería de datos y data masiva [PREDICTWALLSTREET, 2016].

El tercer problema que los inversionistas toman en consideración es la construcción de un portafolio de acciones que permita obtener la más alta rentabilidad con el riesgo más bajo (es decir, reducción de riesgo mediante diversificación de valores) y la inversión solo en sectores industriales que más se conocen (por ejemplo, consumo masivo, minería, pesca, etc.) [DAMODARAN, 1996]. Usualmente, estos trabajos de selección se realizan de manera empírica por los administradores de riesgo y finanzas, lo que implica un esfuerzo excesivo y, en ocasiones, construcción de portafolios deficientes [DAMODARAN, 1996]. En consecuencia, desde la metodología tecnológica, existen trabajos para crear portafolios de acciones tomando en cuenta máquinas de vectores de soporte, procesamiento del lenguaje natural y redes neuronales que han obtenido resultados buenos considerando las preferencias del usuario en términos de riesgo y sectores industriales favoritos [LAI, 2014].

En este sentido, la decisión de la compra y venta de acciones para obtener rentabilidad está ligada a los eventos actuales, noticias de las entidades estudiadas, información actual, desempeño de la economía / negocio, tolerancia de riesgo y sectores de preferencia. En efecto, todas estas variables, información no estructurada y gran cantidad de información hacen de la inversión en bolsa una tarea exhausta, de precisión variable y empírica en muchos casos [DING, 2015]. Por ello, en la actualidad se han considerado algoritmos bajo la teoría de aprendizaje de máquina y minería de datos para simular decisión humana sobre la inversión [LAI, 2014].

Con el desarrollo de la tecnología y computadoras se han desarrollado algoritmos de inteligencia artificial y aprendizaje de máquina para abordar los problemas de la inversión bursátil [DING, 2015].

Diferentes algoritmos han sido estudiados, tales como máquinas de vectores de soporte [CORTES, 1995], redes neuronales [KIMOTO, 1990], NN5 [WONG, 2007], algoritmos genéticos [NEELY, 1997], entre otros. Adicionalmente, una categoría en el dominio de la inteligencia artificial, llamada aprendizaje profundo, ha demostrado ser muy prometedora para simular comportamientos y actividades humanas, tales como jugar GO (AlphaGO), juego de mesa ancestral basado en estrategia e intuición [LASKER, 2011], manejar autos sin conductor, así como lograr una predicción suficientemente precisa y eficiente mediante los avances en procesamiento de datos (uso de GPUs y reducción del costo de almacenaje de datos), como la detección de cáncer [DING, 2015 – GOOGLE, 2015].

En conclusión, se desarrolló una revisión general de los principales problemas para la inversión bursátil: complejidad de elección debido a diferentes fuentes de información masiva y no estructurada, volatilidad del mercado, falta de herramientas tecnológicas que puedan ayudar a una predicción precisa y rápida, y la complejidad para adaptar el riesgo y sectores preferenciales a la elección. Estas cuestiones dan lugar a un problema central, que es la complejidad para la predicción precisa y ágil de movimientos de precios de acciones e índices considerando los perfiles del inversor (es decir, riesgo y sectores) y eventos pasados.

El presente trabajo de fin de carrera hace frente a este problema con la implementación de un algoritmo de aprendizaje profundo en una arquitectura de GPU para la predicción de movimientos de valores tomando en consideración la información actual sobre eventos financieros, resultados y preferencias. Particularmente, se ha demostrado que los eventos financieros procesados bajo aprendizaje profundo han generado muy buenos resultados en términos de rentabilidad, agilidad y precisión [DING, 2014]. El proyecto busca no solo la predicción estática, sino también diaria, y, además de ello, utilizando artículos publicados en portales de noticias en línea apoyado por hardware especializado para obtener mejores resultados de inversión que los trabajos previos en el campo de aprendizaje profundo. El éxito de este proyecto fue medido mediante la rentabilidad promedio obtenida en un año y la precisión de la predicción en comparación con trabajos previos en esta literatura.

2. Marco Teórico

Este apartado detalla los conceptos utilizados en el presente trabajo con el fin de profundizar el entendimiento del mismo, así como de la limitación de ambigüedades. Primero, se discute definiciones sobre el procesamiento natural del lenguaje, necesarias para entender las estructuras que sirven como base para la solución. Segundo, se presentan conceptos sobre la rama de aprendizaje de máquina, lo que ayuda a entender la principal sub-rama en la que se basa la solución: aprendizaje profundo. Tercero, son expuestas definiciones sobre los modelos de predicción de valores considerados en la teoría financiera. Por último, se detalla nociones básicas relacionadas al mercado de valores con el fin de entender mejor la problemática planteada.

2.1. Conceptos generales de procesamiento natural de lenguaje

El procesamiento natural del lenguaje es una rama de la inteligencia artificial, cuyo objetivo principal es dotar a las computadoras de capacidad para procesar tareas que involucran el lenguaje humano (por ejemplo, establecer líneas de comunicación máquina-hombre o simplemente haciendo procesamiento útil de texto o lectura) [JURAFSKY, 2014]. A continuación, se detallan algunos conceptos de esta rama que serán útiles para el presente trabajo:

2.1.1. Análisis sintáctico

Puede definirse como el proceso de asignar descripciones estructurales a las secuencias de palabras en el lenguaje natural. Es decir, este proceso es un análisis formal que se hace a una oración o cadena de palabras para obtener su significado sintáctico y estructura gramatical. De esta manera, un resultado de este proceso es un árbol de análisis sintáctico [JURAFSKY, 2014].

2.1.2. Recuperación de información

Los sistemas de recuperación de información apuntan a proveer procesos para buscar, recuperar y guardar información [STANFORD, 2016]. Estos procesos se sirven de técnicas como el análisis sintáctico o la segmentación de datos para consolidar adecuadamente la información encontrada. Actualmente, con la proliferación de información textual en línea, las tecnologías de recuperación de información se han convertido en tendencia, muchas veces usadas en investigaciones y aplicaciones modernas [SALTON, 1998].

2.1.3. Segmentación de palabras

La segmentación de palabras es una técnica para separar individualmente o en subconjunto las palabras dadas por una entrada textual o hablada. Para idiomas como el inglés, así también lenguas romances, esta separación es trivial porque cada palabra está separada por espacios; sin embargo, para textos en otros idiomas, como el chino, se necesita una descomposición más especializada con muchas más reglas y cuidados [JURAFSKY, 2014].

2.1.4. Eventos estructurados

Los eventos estructurados son la representación de oraciones o eventos de forma estructurada, donde se separan las entidades y los predicados. De esta manera, se capturan las propiedades semánticas de los eventos [MODI, 2016].

Ejemplo:

Oración: “Google presenta AlphaGO”

Evento incrustado: {actor = Google, acción = presenta, objeto = AlphaGO}

2.1.5. Aprendizaje de características

El aprendizaje de características comprende diferentes técnicas para representar datos brutos (e.g. audio, palabras, eventos, imágenes, etc.) en representación que puedan ser usadas en actividades de aprendizaje de máquina, usualmente vectores. [BENGIO, 2013]

2.2. Conceptos generales de aprendizaje de máquina

En este apartado se presentan los conceptos dentro de la rama de aprendizaje de máquina que son útiles para el entendimiento del tratamiento de la problemática mediante la solución propuesta.

2.3. Aprendizaje supervisado

El aprendizaje supervisado es un tipo de entrenamiento para el aprendizaje de máquina en el que se proveen datos etiquetados con el resultado esperado. Es decir, cada ejemplo para entrenar un modelo de aprendizaje de máquina consiste en un par de datos: uno representa la entrada del modelo, y el segundo representa los valores de salida deseados. De esta manera, muchos modelos pueden corregir sus parámetros y afinar sus resultados [MEHRYAR, 2012].

2.3.1. Máquinas de vectores de soporte

Las máquinas de vectores de soporte son modelos de aprendizaje supervisado creados por Vladimir Vapnik. Estas describen hiperplanos de cualquier dimensión (dimensiones mayores a tres), los cuales son usados principalmente para clasificar y hacer regresiones [MEHRYAR, 2012].

El proceso consiste básicamente en la generación de parámetros que describen el hiperplano y que serán afinados durante el entrenamiento mediante datos etiquetados con el fin de presentar mejores resultados [MEHRYAR, 2012].

2.3.2. Redes neuronales artificiales

Las redes neuronales son modelos constituidos en largas arquitecturas de unidades neuronales conectadas, inspirados en cómo los cerebros biológicos resuelven problemas. Típicamente, las redes neuronales consisten en muchas capas de unidades neuronales conectadas (las conexiones son representadas mediante parámetros ajustables) por las cuales fluirá la información de entrada para constituir la salida [BISHOP, 2006]:

El término de redes neuronales agrupa un rango amplio de modelos con características compartidas [BISHOP, 2006]:

- ✓ unidades de procesamiento (inspiradas en las neuronas)
- ✓ conexiones parametrizadas (inspiradas en el proceso de sinapsis)
- ✓ capas y arquitectura (inspiradas en las diferentes agrupaciones de neuronas para diferentes tareas: oler, ver, pensar, dibujar, escribir, etc.)

2.3.3. Modelos de aprendizaje profundo

Los modelos de aprendizaje profundo son un conjunto especial en el cual se intenta obtener resultados sólidos y abstracciones de alto nivel mediante el uso de múltiples

capas de procesamiento y entrenamiento a través de volúmenes masivos de datos [BENGIO, 2015].

Así, este tipo de modelos son una extensión de complejidad de los modelos de redes neuronales clásicos. Esta complejidad adicional requiere de consideraciones especiales como el uso de las GPU para acelerar el entrenamiento o el menor costo de almacenamiento de datos que han hecho posible la factibilidad de estos modelos [DENG, 204].

2.3.3.1. Redes recurrentes profundas y Memoria de largo / corto plazo (LSTM)

Las redes neuronales recurrentes son una extensión de las redes multicapas en donde el resultado de cada nodo sirve como entrada del nodo siguiente, así, permite conexiones cíclicas que son importantes cuando se considera series de tiempo históricas. Así, mientras que la extensión cíclica de las redes recurrentes sobre las redes multicapas parece trivial, las implicancias de esta son mayores, es decir, mientras que las redes multicapa permiten mapear la entrada y salida, las redes recurrentes permiten mapear la histórica completa de los ejemplos mediante esa extensión cíclica [GRAVES, 2008].

Un problema de estas redes es el problema de desvanecimiento de gradiente, es decir, que para el entrenamiento de este tipo de redes se propaga el error hacia atrás a través del tiempo, cada vez el factor va reduciendo el efecto de este error. Así, mientras más profunda sea la capa de actualizar, más lento será su entrenamiento, llegando incluso a ser imposible entrenarla en tiempos razonables. En consecuencia, se ha desarrollado la memoria de largo-corto plazo (LSTM) para poder conllevar este problema, añadiendo la transición de estados entre nodos que permite mantener el impacto del error mientras se propaga hacia atrás en una suerte de “canal” especial llamadas compuertas [GRAVES, 2008].

2.3.3.2. Redes neuronales tensoriales

Este tipo de redes neuronales fue introducido por un grupo de investigadores de procesamiento de lenguaje natural de la universidad de Stanford con el fin de presentar una alternativa de solución para la falta de datos completos en una base de conocimiento. De este modo, el proceso principal a solucionar es de qué manera se pueden representar eventos mediante vectores para poder mantener su estructura semántica, es decir, vectores que puedan tomar en cuenta el orden de las palabras. Para ello, se presentó la arquitectura de redes neuronales tensoriales, en la cual se reemplaza los pesos de las redes neuronales multicapas por tensores, así, después de un entrenamiento poder capturar esas representaciones de eventos y sus relaciones mediante los valores de esos tensores [SOCHER, 2013].

2.3.4. Indicadores de desempeño

2.3.4.1. Coeficiente de correlación de Matthews

Es un coeficiente que sirve para medir la calidad de clasificaciones binarias, fue introducido por Brian W. Matthews en 1975. La ecuación del presente coeficiente es [MATTHEWS, 1975]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Ecuación 1: Coeficiente de correlación de Matthews

- ✓ TP Positivos verdaderos
- ✓ FP Falsos positivos
- ✓ FN Falsos negativos
- ✓ TN Negativos verdaderos

2.3.4.2. Rentabilidad y estrategias de inversión

La rentabilidad se define como la diferencia positiva entre la venta y costo de un producto, en este contexto el término recae sobre la ganancia generada por la diferencia de precios generada entre la compra y venta de una acción o índice bursátil [LAVRENKO, 2008].

Una estrategia de inversión son un conjunto de reglas que están diseñadas para obtener algún retorno positivo. [LAVRENKO, 2008].

2.3.5.GPU

La GPU, unidad de procesamiento gráfico, es un circuito especializado diseñado para procesar instrucciones de cómputo intensivo de manera más eficiente que una CPU. Inicialmente creado con el fin de soportar videojuegos o procesamiento de imágenes/video, en los últimos años es usada por investigadores y académicos para hacer mejoras innovadoras y más eficientes en el entrenamiento de redes neuronales artificiales complejas [TENSORFLOW, 2016]. En consecuencia, modelos de aprendizaje profundo que tenían como principal limitación el cómputo intensivo han llegado a ser factibles mediante el uso de este dispositivo [NVIDIA, 2016].

2.4. Conceptos generales en la teoría de la predicción de valores

2.4.1.Hipótesis del caminante aleatorio

La hipótesis del caminante aleatorio es una teoría que indica que la evolución de precios de acciones en un mercado determinado evoluciona de acuerdo a un conjunto de pequeños movimientos aleatorios. Como consecuencia de esta teoría, las tendencias pasadas o datos históricos no pueden ser usados, y la mejor variable para hallar el precio futuro es el precio de hoy [BURTON, 2007].

En contraste con esta hipótesis, académicos y teóricos (como Martin Weber) han demostrado que los precios futuros siguen tendencias pasadas en muchos casos, por lo que generalizar la hipótesis no concuerda con lo estudiado empíricamente [BURTON, 2007].

2.4.2.Hipótesis del mercado informacionalmente eficiente

La hipótesis del mercado informacionalmente eficiente fue desarrollada por el profesor Eugene Fama, esta teoría es ampliamente conocida e indica que el precio de cada acción es resultado de la información disponible. Por lo tanto, cualquier cambio en el precio es resultado de nueva información disponible [HARDER, 2010].

Bajo esta teoría existen consideraciones como el tiempo de propagación de la información y el tiempo de procesamiento. Así, mientras la información se extienda más

rápido y mientras sea procesada por más personas/modelos el nuevo precio será reflejado más rápido [HARDER, 2010].

2.5. Conceptos generales mercado de valores

Las siguientes definiciones fueron tomadas del libro de *Rodney Hobson Shares Made Simple: A Beginner's Guide to the Stock Market* [HOBSON, 2012].

2.5.1. Acción

Una acción es un título emitido por una entidad que representa una alícuota del capital total de modo que un accionista pueda conferir derechos de voto. Este título representa un instrumento financiero porque se invierte su adquisición esperando ganancias mediante los dividendos o cambios positivos en el precio.

2.5.2. Mercado de valores

El mercado de valores es un conjunto de compradores y vendedores certificados que comercian acciones de diferentes entidades. Los más conocidos son *Dow Jones*, *National Association of Securities Dealers Automated Quotation* (NASDAQ), Londres SEG, etc.

2.5.3. Volatilidad

La volatilidad es el grado de variación que tiene un precio sobre el tiempo: mientras mayor sea la volatilidad, mayor será el riesgo asumido y mayor será la ganancia (o pérdida, en su defecto) esperada.

2.5.4. Liquidez

La liquidez es la característica de un activo (i.e. instrumento financiero) de ser convertido en dinero efectivo mediante su venta.

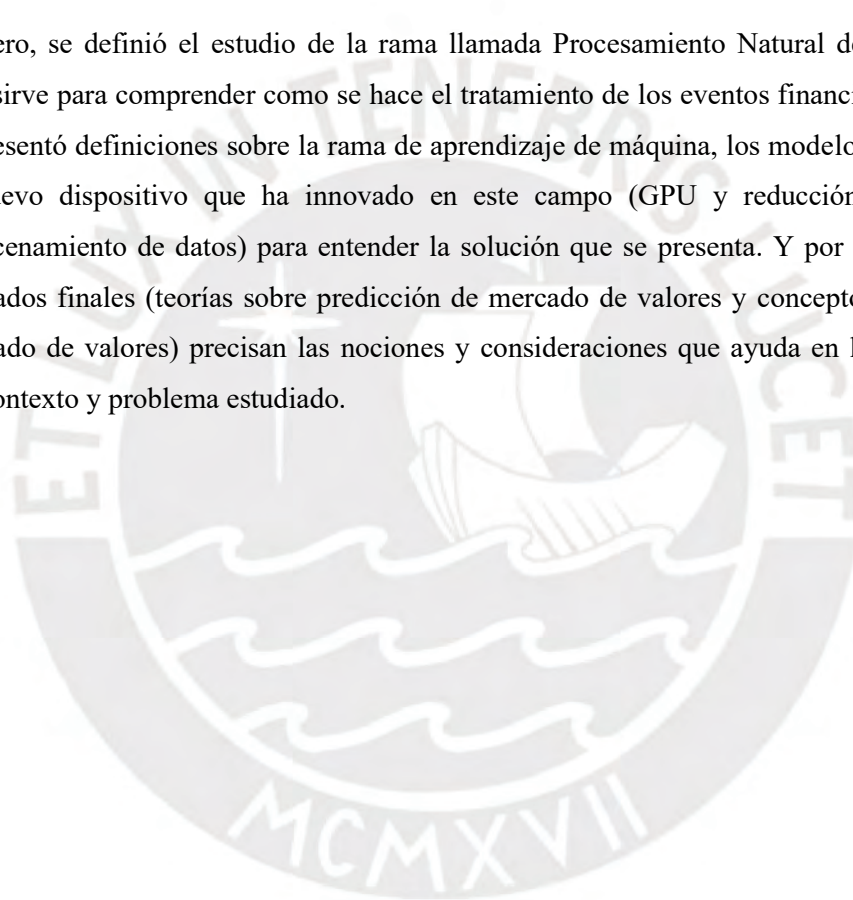
2.5.5. Cotización

Una cotización es la tasación que se da a un activo para su comercialización.

2.6. Conclusión

En conclusión, en la presente sección se han detallado los conceptos necesarios para el entendimiento y limitación del trabajo.

Primero, se definió el estudio de la rama llamada Procesamiento Natural del Lenguaje, el cual sirve para comprender como se hace el tratamiento de los eventos financieros. Segundo, se presentó definiciones sobre la rama de aprendizaje de máquina, los modelos disponibles y el nuevo dispositivo que ha innovado en este campo (GPU y reducción de costes en almacenamiento de datos) para entender la solución que se presenta. Y por último, los dos apartados finales (teorías sobre predicción de mercado de valores y conceptos generales de mercado de valores) precisan las nociones y consideraciones que ayuda en la comprensión del contexto y problema estudiado.



3. Estado del arte.

El problema de la inversión y predicción sobre las acciones ha sido una actividad de alta rentabilidad y es consecuencia de la necesidad corporativa de adquirir liquidez de forma rápida y medida. Debido a la importancia, rentabilidad y atractivo de este tipo de inversiones, se han desarrollado diferentes métodos, productos e investigación con el fin de obtener los mejores resultados.

3.1. Métodos manuales

Existen dos métodos manuales para predecir el comportamiento de la bolsa bursátil, así, estas son el análisis fundamental y el análisis técnico.

3.1.1. Análisis fundamental

En el análisis fundamental, el inversor trata de obtener beneficios bursátiles al conocer las causas internas y externas de las compañías que generan cambios en su valor [GRIN, 2009]. En las causas internas, se ha creado una teoría extensa sobre este tipo de análisis para consolidar el conocimiento adquirido. Esta teoría contempla el uso de ratios e índices para sintetizar y medir los resultados y desempeño de las entidades (como ratio de solvencia, ratio de rentabilidad sobre inversión, etc.). Por otro lado, considerando las causas externas, se toman en cuenta los eventos, acontecimientos, cambios organizacionales y noticias para evaluar cómo las fuerzas externas pueden impactar en el desempeño de las compañías valoradas. Este método requiere un alto grado de especialización, así como un esfuerzo exhaustivo por la gran cantidad de datos financieros, eventos e información no estructurada [GRIN, 2009].

3.1.2. Análisis Técnico

En el análisis técnico, el inversor trata de obtener beneficios bursátiles extrapolando los patrones encontrados en análisis previos [GRIN, 2009]. Es decir, usa herramientas gráficas para visualizar datos históricos recolectados y tratar de encontrar patrones útiles que puedan estar sucediendo. Si la predicción de la tendencia es la correcta, obtendrá beneficios. Este análisis es limitado debido a que solamente examina datos, sin comprender el significado y causas que originan esos patrones [GRIN, 2009].

3.2. Proyectos de investigación para la predicción de valores futuros

En este punto, para hacer frente al problema detallado líneas arriba, se revisan los trabajos con resultados destacados sobre implementaciones tecnológicas con foco en inteligencia artificial en resolución de problemas de predicción bursátil.

3.2.1. Análisis de series de tiempo para la predicción de valores futuros

Estas implementaciones tecnológicas utilizan una perspectiva estadística para poder evaluar los datos históricos y gráficas obtenidas de manera automática. Se utilizan diversos enfoques, como, por ejemplo, promedio móvil autorregresivo integrado [BOX, 1976] o regresión multivariable [PESARAN, 1994]. Estos han sido inspirados en el análisis técnico, y se les ha dado un enfoque estadístico para tomar ventaja de la fuerza computacional; por lo tanto, heredan la limitación más importante que es la falta de conocimiento de las causas y entendimiento del significado de los resultados más allá del modelo numérico [LAI, 2014].

3.2.2. Métodos de vectores de soporte de máquina para la predicción de valores futuros

En las últimas décadas, con el resurgimiento y desarrollo de la inteligencia artificial, así como del progreso de los recursos computacionales, se ha atacado el problema descrito mediante diferentes modelos. Entre los más promisorios se encuentran las máquinas de vectores de soporte.

En estos modelos se reconocen y analizan patrones mediante la clasificación de los conjuntos de datos. Se han realizado muchos estudios sobre este enfoque, como el trabajo de Cao y Tay en el 2001, para predecir los movimientos del índice S&P 500 [CAO, 2001] y el trabajo en la bolsa de valores de Corea del Sur por Kim Kyoung [KIM, 2003]. Estos trabajos han obtenido resultados con un mejor estimado de precisión de 57.8 %, lo cual representa una mejora significativa frente a los trabajos usando series de tiempo.

Las principales limitaciones en estos trabajos siguen siendo la falta de evaluación de eventos, noticias, y factores externos que no solo implican resultados, sino también causas.

3.2.3. Redes neuronales para la predicción de valores futuros

Las redes neuronales son modelos de inteligencia artificial inspirados en las funciones biológicas del cerebro, pues son usadas para simular actividades consideradas exclusivas de los humanos [BENGIO, 2015]. Desde décadas pasadas se han implementado diferentes redes neuronales para la predicción de valores bursátiles. Por ejemplo, el trabajo titulado “Predicción en el mercado de valores usando redes neuronales” implementa una red neuronal de tres capas para poder obtener patrones bursátiles y, así, obtiene resultados bastante sólidos, sin embargo, con un tiempo de procesamiento prohibitivo para la alta volatilidad del mercado de valores [KIMOTO, 1990]. Así mismo, el trabajo “Usando redes neuronales para predecir precios de acciones” concluye que la aplicación de redes neuronales ofrece mejores resultados que los métodos estadísticos y de regresión, aunque existen problemas a ser solucionados, como el tiempo y variaciones no capturadas por un sistema caótico como la bolsa de valores [LAWRENCE, 1997].

Por otro lado, las investigaciones sobre la predicción bursátil mediante el uso de procesamiento natural del lenguaje llegan con Lavrenko en su trabajo de “Mining of concurrent text and time series” en el cual utilizan el método de bolsa de palabras para la representación vectorial de noticias y los resultados de precisión se tallan en 56.38% con un MCC de 0.07 [LAVRENKO, 2000].

En los últimos años, el aprendizaje profundo en redes neuronales junto con las nuevas innovaciones tecnológicas como las GPU para aprendizaje de máquina y la reducción de costo de almacén de datos han permitido obtener resultados precisos con una velocidad superior [TENSORFLOW, 2016]. Así, por ejemplo, el trabajo sobre el aprendizaje profundo dirigido por eventos para la predicción ha demostrado en promedio una rentabilidad del 6 % (igual o superior al índice de S&P al que fue aplicado) [DING, 2014].

3.3. Productos comerciales

En este apartado se detalla dos implementaciones comerciales con bastante poder de predicción [PREDICTWALLSTREET, 2016] [STOCK FORECASTING, 2016].

3.3.1. Predict Wall Street

Esta herramienta tecnológica permite la predicción de los valores futuros de la bolsa de valores sobre la base de trabajos realizados por el Dr. Craig. Así, permite evaluar la precisión con que se ha ido desarrollando y los niveles de riesgo aceptados en cada caso. Como explica, existen dos etapas para la predicción de valores. Primero, se colecta información sobre publicaciones de analistas financieros e inversores para entender lo que está sucediendo en esos momentos. Segundo, se aplican algoritmos de minería de datos para poder explotar la información recolectada y encontrar patrones para hacer predicciones sobre el comportamiento de la bolsa bursátil. Sus resultados son muy diversos, con precisión de hasta un 97 % en algunas empresas [PREDICTWALLSTREET, 2016].

3.3.2. Stock Forecasting

El servicio ofrece predicciones sobre la bolsa a diario, corto y largo plazo, este usa algoritmos de inteligencia artificial y métodos estadísticos para el procesamiento de sus resultados. Se logran predicciones muy precisas sobre el tiempo; sin embargo, la rentabilidad ligada a las predicciones están igual o levemente por encima del desempeño de los índices bursátiles (por ejemplo, S&P 500) [STOCK FORECASTING, 2016].

3.4. Conclusiones

Como se ha detallado, en años pasados los analistas financieros e inversores utilizaban dos métodos bien definidos: método de análisis técnico y método de análisis fundamental. Por un lado, el primero está limitado por la falta de comprensión de las causas y fuerzas que impactan en el movimiento de los resultados y la creación de patrones. Por otro lado, el segundo está limitado fuertemente por la subjetividad del analista y también por la información masiva y no estructurada que existe.

Posteriormente, debido al progreso tecnológico de los ordenadores, se han implementado diferentes herramientas y algoritmos, para la simulación y predicción de valores bursátiles. Esta nueva perspectiva ofrece resultados prometedores en ámbitos fundamentales como

rapidez de procesamiento, precisión y reducción de complejidad, aunque con dificultad para combinar todos estos.

En la actualidad, la llegada de innovaciones tecnológicas mencionadas en apartados anteriores, como la de las GPU para el cálculo numérico y la reducción del costo en el almacenamiento de datos, ha permitido que modelos alimentados con datos masivos, como el aprendizaje profundo, ataquen este problema y obtengan resultados prometedores en un tiempo menor.



4. Objetivo general

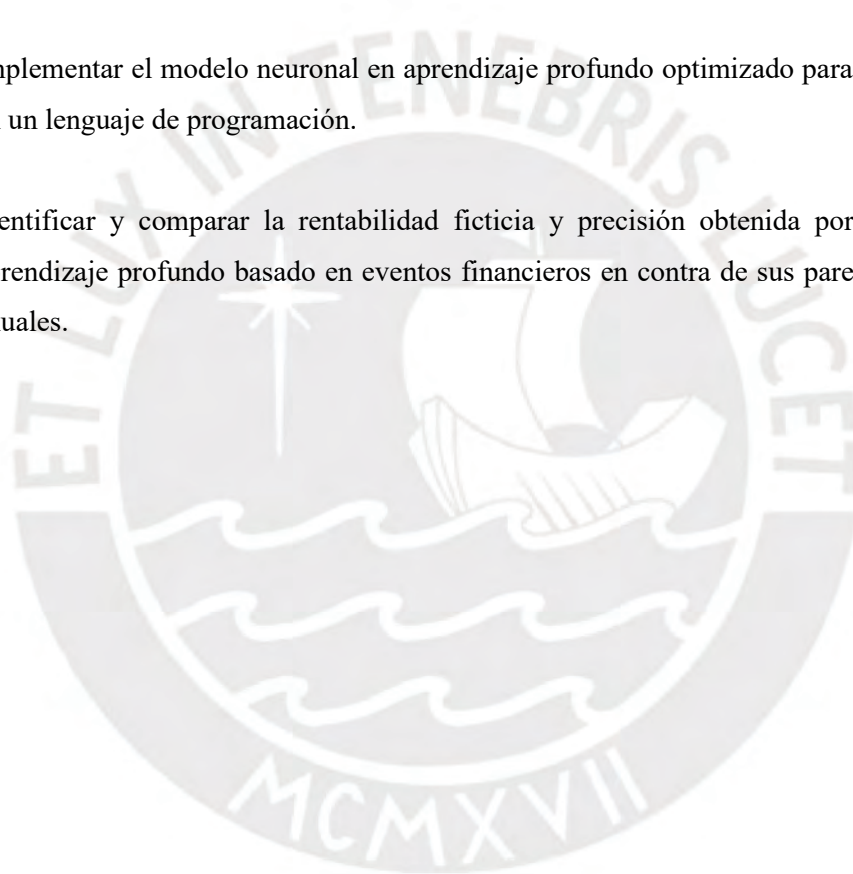
El objetivo general es diseñar e implementar un modelo integrado de red neuronal profunda para la predicción del índice bursátil S&P 500 mediante el análisis estructurado de eventos, noticias y resultados financieros corporativos extraídos de la web.



5. Objetivos específicos

Los objetivos específicos de este proyecto de fin de carrera se presentan a continuación:

1. Diseñar un modelo de red neuronal en aprendizaje profundo basado en eventos financieros como opción de solución al problema predicción de valores bursátiles
2. Integrar herramientas para recolectar y procesar eventos financieros y generales para alimentar el algoritmo de aprendizaje profundo
3. Implementar el modelo neuronal en aprendizaje profundo optimizado para el uso de GPU en un lenguaje de programación.
4. Identificar y comparar la rentabilidad ficticia y precisión obtenida por el modelo de aprendizaje profundo basado en eventos financieros en contra de sus pares en escenarios anuales.



6. Resultados esperados

Para el objetivo #1:

- ✓ Arquitectura de la red neuronal
- ✓ Diseño de los tipos de red neuronal

Para el objetivo #2:

- ✓ Módulo de integración con una herramienta de extracción de información
- ✓ Información estructurada para entrenar la red neuronal en texto plano
- ✓ Información vectorizada de palabras y eventos financieros

Para el objetivo #3:

- ✓ Red neuronal de aprendizaje profundo implementado contenido en un componente reutilizable
- ✓ Componente implementado para el entrenamiento y obtención de resultados de la red neuronal

Para el objetivo #4:

- ✓ Análisis de rentabilidad neta anualizada utilizando los precios históricos de S&P 500 versus los predichos y una estrategia de inversión básica.
- ✓ Análisis de la experimentación para la obtención de la precisión y el coeficiente de correlación de Matthews.

7. Herramientas y metodología

El detalle de las herramientas, métodos y procedimientos en relación a los resultados esperados se presenta a continuación:

Tabla 1: Resultados esperados

Resultado Esperado	Herramienta / método
<ul style="list-style-type: none"> - RE1: Arquitectura de la red neuronal - RE2: Pseudocódigo de la red neuronal a usar 	<ul style="list-style-type: none"> - Microsoft Visio - Deeplearning4J
<ul style="list-style-type: none"> - RE3: Módulo de integración con una herramienta de extracción de información - RE4: Información estructurada para entrenar la red neuronal en texto plano - RE5: Información vectorizada de palabras y eventos financieros 	<ul style="list-style-type: none"> - Python - Java - IntelliJ - Stanford Extractor de Información - OpenCSV - Word2Vec - Red neuronal tensorial - Scrapy
<ul style="list-style-type: none"> - RE6: Red neuronal de aprendizaje profundo implementado contenido en un componente reutilizable - RE7: Componente implementado para el entrenamiento y obtención de resultados de la red neuronal 	<ul style="list-style-type: none"> - Python - Java - IntelliJ - Deeplearning4J - GPU Nvidia Grid
<ul style="list-style-type: none"> - RE8: Análisis de rentabilidad neta anualizada utilizando los precios históricos de S&P 500 versus los predichos y una estrategia de inversión básica. . - RE9: Análisis de la experimentación para la obtención de la precisión y el coeficiente de correlación de Matthews 	<ul style="list-style-type: none"> - Microsoft Excel - Estrategia de inversión básica - Análisis de rentabilidad neta anualizada - Coeficiente de correlación de Matthews - Precisión de pronóstico

7.1. Herramientas

En este apartado describimos cada una de las herramientas mencionadas.

7.1.1.IntelliJ

IntelliJ es un IDE que ofrece editores de código, analizadores, depuradores, entre otros, y que permite un desarrollo rápido para aplicaciones móviles, de escritorio y aplicaciones web. Soporte para Java e integración con otros lenguajes como Python. [INTELLIJ, 2016].

La elección de este IDE se basa en tres puntos. Primero, es compatible con los lenguajes seleccionados: Java y Python. Segundo, esta herramienta simplifica muchos aspectos claves como el autocompletado, sangría automática, re-factorización, entre otros. Por último, ofrece integración completa con las librerías seleccionadas: deepLearning4J y NLPCore de esta manera la herramienta será soporte para la tarea de implementación de la red neuronal.

7.1.2.Python

Python es un lenguaje de programación multiparadigma, altamente legible, interpretado y de propósito general [PYTHON, 2106].

La razón fundamental de la elección de este lenguaje es la integración usada con la librería TensorFlow. Este lenguaje será usado para la implementación de las redes neuronales, el módulo de integración y la interfaz visual.

7.1.3.Deeplearning4J

Es una librería de código abierto para la implementación de redes neuronales en Java. La flexibilidad de esta librería permite el uso de GPU [DEEPLEARNING4J, 2016].

La elección de esta librería se da por dos razones: Primero, otorga compatibilidad para el uso de GPU. Y segundo, soporta modelos de aprendizaje profundo. Con ayuda de esta librería se implementará la red neuronal de aprendizaje profundo propuesta.

7.1.4. GPU Nvidia Grid – Arquitectura Kepler

Es una arquitectura desarrollada por la empresa NVIDIA, sucesora de la arquitectura Fermi. Su enfoque general es mejorar la eficiencia, flexibilidad y desempeño en instrucciones que requieren cómputo intensivo [NVIDIA, 2016].

La razón fundamental de su elección es que esta tecnología hará mucho más eficiente el entrenamiento de la solución versus un entrenamiento soportado por CPU.

7.1.5. Stanford Extractor de información abierto (Open Information Extractor)

Es un aplicativo que ayuda a la extracción de información desde diferentes fuentes y las agrupa en tuplas relacionadas [STANFORD, 2016]

La elección fue hecha por dos razones: Primero, es de código abierto, lo cual ayuda a la personalización del mismo para diferentes propósitos. Y segundo, para la información extraída puede ser almacenada directamente en forma de tuplas – lo cual automatiza el proceso de estructuración. Esta aplicación realizará la extracción de información de las páginas web consideradas.

7.1.6. Word2Vec

Es una red pre-entrenada creada bajo el auspicio de Google DeepMind para la representación vectorial de palabras [MIKOLOV, 2013].

La razón fundamental por la que se eligió está red neuronal pre-entrenada es porque ofrece un novedoso y eficiente método para transformar palabras en vectores.

7.1.7. Microsoft Excel

Es una aplicación distribuida como parte de la suite de oficina Microsoft Office. Excel tiene como características principales el manejo de hojas de cálculo, transformación de datos, aplicación de fórmulas, visualización mediante gráficos, paquetes de funciones (como paquete estadístico, matemático, financiero, etc.), entre otros [EXCEL, 2016].

Debido a la sencillez de su uso e inclusión de paquetes estadísticos, esta herramienta servirá para realizar la experimentación numérica y el informe de análisis de rentabilidad neta anualizada.

7.1.8. Java

Es un lenguaje de propósito general orientado a objetos diseñado para correr en una máquina virtual y de este modo ser multiplataforma [JAVA, 2017]

La principal razón por la que se utiliza este lenguaje es por su compatibilidad con DeepLearning4J, que fue la librería escogida para la realización de la parte fundamental del presente trabajo.

7.1.9. Microsoft Visio

Esta herramienta de la suite de Microsoft Office permite la diagramación de diferentes flujos, gráficos, etc. [OFFICE, 2017].

Esta herramienta fue escogida para la diagramación de las diferentes arquitecturas y flujos que se presentarán en este proyecto.

7.1.10. OpenCSV

Es una librería muy simple para el manejo de archivos CSV implementada en Java [OPENCsv, 2017].

Esta librería fue elegida por su compatibilidad con el lenguaje de programación escogido y por la necesidad de manejar archivos CSV para los entrenamientos de las redes neuronales propuestas.

7.1.11. Scrapy

Es un entorno de trabajo de código abierto y colaborativo que permite extraer información desde diferentes sitios web de una manera fácil y escalable [SCRAPY, 2017].

Esta herramienta se escogió para la recolección de información desde los portales web que servirán como entrada para las redes neuronales propuestas.

7.2. Métodos

A continuación se presentarán los métodos usados en el presente proyecto.

7.2.1. Precisión de pronóstico

Se define como precisión al ratio entre los ejemplos correctamente pronosticados versus los ejemplos totales [POWERS, 2011]. De esta manera, este indicador sirve para poder medir la capacidad de precisión del presente trabajo.

7.2.2. Coeficiente de correlación Matthews

Es un coeficiente que sirve para medir la calidad de clasificaciones binarias, fue introducido por Brian W. Matthews en 1975. Este coeficiente toma en cuenta todos los aspectos de la matriz de confusión: falsos positivos, falsos negativos, positivos verdaderos y negativos verdaderos [MATTHEWS, 1975].

El presente trabajo hace uso de este coeficiente por la consideración que tiene en la clasificación binaria (dos clases) de todos los aspectos de evaluación y no solo de los casos de éxito.

7.2.3. Estrategia de inversión

Una estrategia de inversión es un conjunto de reglas que están diseñadas para obtener algún retorno positivo. La estrategia de inversión más simple y que es tomada en muchos proyectos de esta naturaleza es del siguiente modo [LAVRENKO, 2008]:

- ✓ Durante el día de simulación si el inversor sabe que el índice subirá, él compra el índice al precio al que abre y si durante ese día sube un 2% lo vende inmediatamente, sino lo vende al precio de cierre.
- ✓ Durante el día de simulación si el inversor sabe que el índice bajará, él usará el método *shorting*, es decir, se prestará el índice al inicio del día y lo venderá al precio al que abre, y si durante el día baja 1% comprará la misma cantidad de acciones, en caso contrario, comprará con el precio de cierre para hacer el encubrimiento.
- ✓ Debido a que las comisiones varían mucho de acuerdo a la cantidad intercambiada y dependiendo del bróker, no se considera el uso de comisiones en la simulación.

7.2.4. Análisis de ganancia neta anualizada

Este análisis se compone de un conjunto de métricas para determinar el monto promedio de retorno que genera un portafolio de inversiones, así, determinar qué portafolio es más conveniente y cuál es el riesgo asociado [BRINSON, 1995].

Esta metodología se usa para medir el desempeño sobre la rentabilidad en el informe de rentabilidad neta anualizada.

7.3. Metodología y plan de trabajo

Para la correcta culminación del presente proyecto se ha seguido la siguiente estructura:

7.3.1. Estudio de la literatura

La primera fase del presente trabajo es el estudio del estado del arte y las tecnologías sobre predicción de valores bursátiles con distintos tipos de métodos: estadísticos, métodos tradicionales y, con más detalle por el alcance del proyecto, métodos en la rama de aprendizaje de máquina. Así mismo, puesto que el proyecto está enfocado en utilizar las últimas tendencias tecnológicas, como aprendizaje profundo, Word2Vec, *Skip-Gram* y las ramas ya mencionadas, el objetivo del mismo es ser comparado y evaluado contra estos resultados de estado del arte.

7.3.2. Extracción y estructuración de la información

En esta fase se crea la base de datos que debe ser pre-procesada y sirve como entrada para la red neuronal. En detalle, la información recolectada consiste en los títulos de noticias en línea, precios y tendencias de la bolsa bursátil. Siguiendo esta línea, la información obtenida se preprocesa y estructura, de manera que se obtienen eventos estructurados.

7.3.3. Representación vectorial de los eventos

En esta fase se representa la información estructurada mediante vectores que son la entrada directa a la red neuronal propuesta. Esta fase se divide en dos actividades principales. Primero, se vectorizan las palabras individuales mediante el algoritmo de *Skip-Gram*, que permite capturar el significado individual de las palabras. Segundo, se

vectorizan los eventos mediante el modelo de redes neuronales tensoriales que permiten capturar la relación entre palabras dentro de un evento.

7.3.4. Diseño e implementación de la red neuronal propuesta

En esta fase central del proyecto se propone un modelo de red recurrente neuronal profunda con LSTM que permite pronosticar el movimiento de la bolsa bursátil en el corto, mediano y largo plazo (365 nodos de entrada representa un año completo) con una capa escondida. De esta manera, se consideran los eventos financieros de un año para poder pronosticar el movimiento bursátil del día siguiente (con las clases +1,-1).

7.3.5. Entrenamiento

La red central propuesta es una red de aprendizaje profundo por lo cual el entrenamiento se desarrollará en mediante la ayuda de un GPU para reducir el tiempo de entrenamiento, así, en esta fase la red neuronal recibe el conjunto de representaciones vectoriales obtenidos de la red anterior (vectorización de eventos) de acuerdo a los plazos, es decir, cada nodo de la red neuronal recibe un vector de 300 características que representa a los eventos del día, así se introducirán 365 días (el año histórico) para poder predecir el movimiento del día siguiente. Solo se actualizan los pesos si el pronóstico de movimiento no es correcto mediante retroalimentación a través del tiempo.

7.3.6. Evaluación de la red neuronal propuesta

El presente trabajo presenta dos facetas. Primero, es necesaria la evaluación en términos técnicos de entrenamiento de redes neuronales, por lo que se utilizan dos medidas para tallar el desempeño de esta evaluación: MCC y precisión. Segundo, es necesaria la evaluación en términos financieros; por lo tanto, se mide la cantidad de rentabilidad que es capaz de generar con una estrategia básica de comercio de valores en el plazo de un año con un capital inicial.

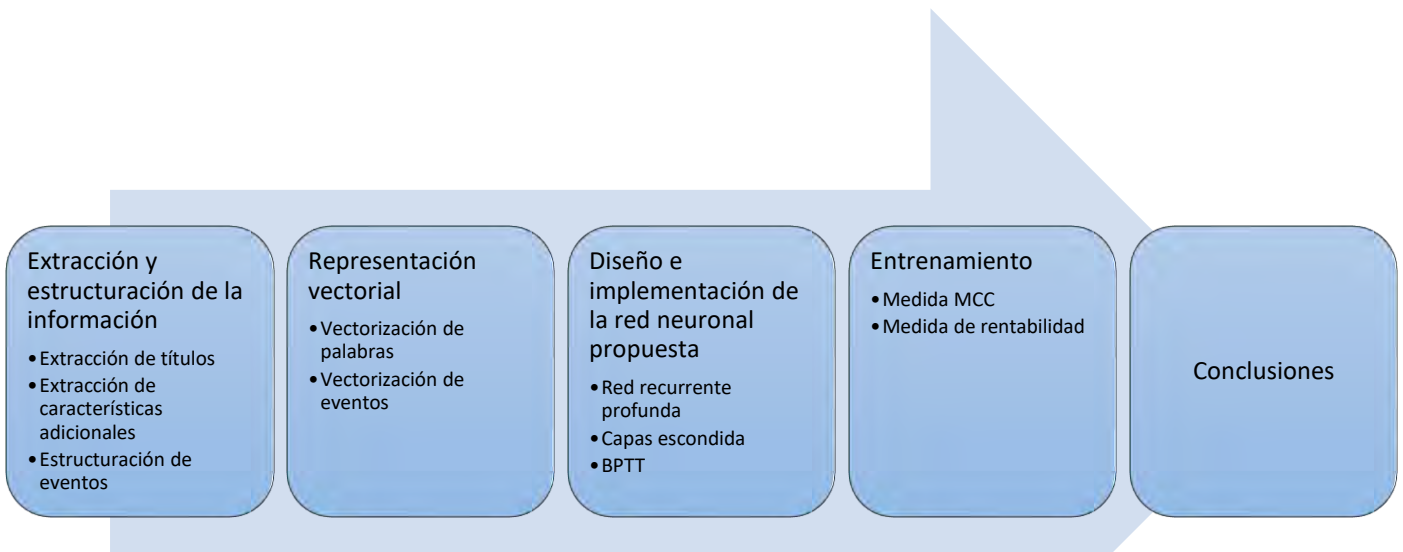


Imagen 1: Flujo global

7.3.7. Conclusiones

En esta etapa final se detallan las observaciones obtenidas, es decir, los resultados finales y comparados, el análisis sobre la eficiencia y las extensiones que pueden servir para mejorar el resultado.

8. Alcance, limitaciones y riesgos

El presente trabajo es una alternativa de solución al problema de la predicción de los precios de acciones en el mercado bursátil, específicamente para la experimentación se somete el índice de S&P 500 a evaluación. En detalle, la solución muestra la mejor estimación para cada acción para saber, así, cuándo es factible la venta o compra del índice con el fin de obtener ganancias o, reducir pérdidas y riesgos. Este objetivo, la predicción de los precios de acciones, se logra mediante la extensión tecnológica del método fundamentalista. Es decir, se estructura y analiza datos masivos de publicaciones financieras (como noticias, resultados financieros, opinión de analistas, etc.) y, sobre la base de estas, se pronostica los futuros precios en una ventana de tiempo predefinida.

Así, esta solución es diseñada e implementada bajo la teoría de redes neuronales de aprendizaje profundo. Primero, se define las estructuras de datos, arquitectura y diseño que soporten las redes neuronales a implementar. Segundo, se adaptan los módulos de extracción y estructuración de la información junto con la implementación completa de la red neuronal de aprendizaje profundo orientada al uso de GPU. Por último, se analiza el desempeño del presente trabajo en las dos facetas de este proyecto: análisis de renta anualizada y el análisis de la precisión y coeficiente de correlación Matthews. Esta solución busca mejorar el desempeño temporal y la precisión que mostraron los resultados de trabajos similares.

Es clave precisar, que el presente trabajo considera ejecuciones y pronósticos en el ámbito del índice S&P 500 de la bolsa de New York, es decir, con la misma divisa de transacción, así mismo, solo se toma fuentes de información en solo un lenguaje en particular - inglés.

8.1. Limitaciones

Se listan las siguientes limitaciones consideradas:

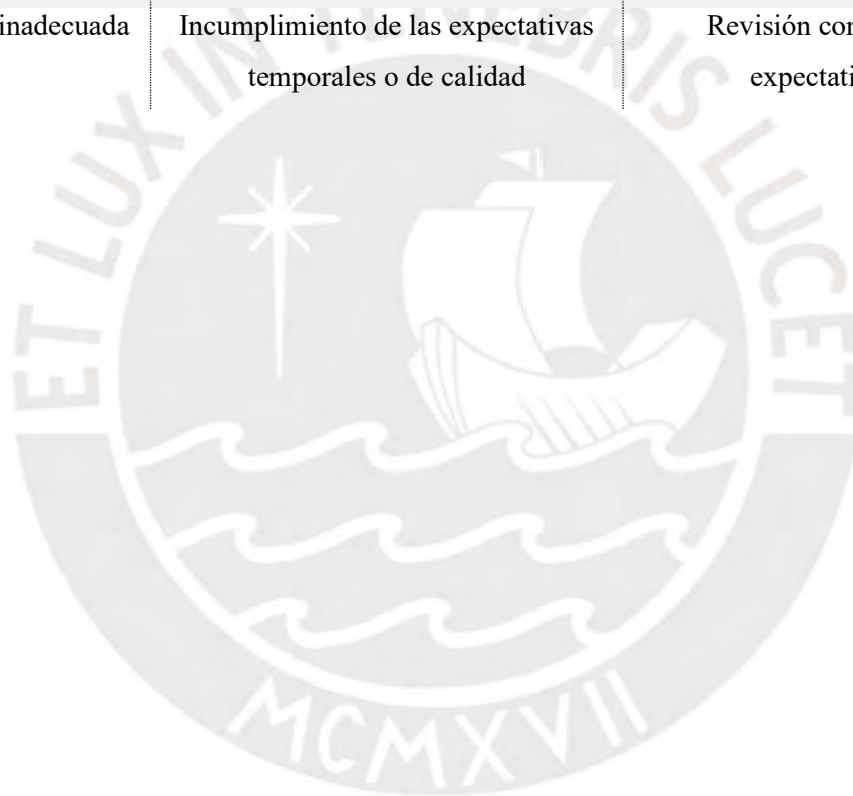
- ✓ Los datos masivos usados como entrada (aproximadamente 100GB de información), así como los precios históricos de las acciones, limitan a que el ámbito de ejecución sea solo en bolsas similares, es decir, de la misma divisa de transacción y con fuentes de información en el mismo lenguaje.
- ✓ Dada la complejidad y la naturaleza estocástica de las redes neuronales a implementar (aprendizaje profundo) el tiempo de entrenamiento y verificación varía de acuerdo al hardware y software del entorno de ejecución, por ello, todas las ejecuciones, entrenamientos y validaciones se realizarán con los mismos recursos informáticos.

8.2. Riesgos

Los riesgos identificados se detallan a continuación.

Tabla 2: Gestión de riesgos

Riesgos identificados	Impacto	Plan de mitigación
Entorno de ejecución no disponible	Imposibilidad de pruebas en el ámbito definido (i.e. uso de GPU)	Considerar servicios en la nube que provean entornos de ejecución en aprendizaje de máquina
Planificación inadecuada	Incumplimiento de las expectativas temporales o de calidad	Revisión continua del avance y expectativas del proyecto



9. Justificación y viabilidad

A continuación, se detalla la justificación del presente trabajo.

9.1. Justificación del proyecto de fin de carrera

Las acciones bursátiles se han convertido en un instrumento financiero altamente rentable y clave para la operación de las grandes transnacionales que cotizan diariamente su precio en la bolsa. En consecuencia, una predicción adecuada de los precios ayudaría a generar ganancias altamente rentables sobre la inversión hecha. De este modo, se justifica la importancia del problema sobre la predicción futura de precios de las acciones en el mundo actual.

Por otro lado, actualmente es posible encontrar grandes cantidades de información (incluso sobre los precios y cotizaciones de la bolsa) disponibles en la web, ya sea en portales noticieros o páginas especializadas en finanzas corporativas. Es clara, entonces, la importancia de la extracción, consolidación y estructuración adecuada de toda esta información masiva para su correcta utilización y explotación. Es decir, existe un área de oportunidad grande sobre la utilización eficiente de toda esta información disponible.

Actualmente, con la evolución acelerada de las tecnologías, se han desarrollado dos factores que han sido claves para hacer posible la ejecución de implementaciones desarrolladas bajo la teoría de aprendizaje profundo: la disminución del costo de almacenamiento de datos y el uso de GPU para el entrenamiento de redes neuronales. Así, en los últimos días, las redes neuronales en aprendizaje profundo han demostrado su potencial mostrando resultados excepcionales y sorprendentes para simular actividades humanas con un alto grado de precisión y rendimiento temporal. He ahí la importancia de usar esta perspectiva para abordar el problema en cuestión. El presente trabajo considera, entonces, esta tecnología viable e innovadora, puesto que en años anteriores no existía el hardware necesario o era impermisible en términos económicos. Adicionalmente, la relevancia del problema del pronóstico de los precios de acciones y la gestión adecuada de la cantidad de datos masivos disponibles en la web, ha sido justificada en los párrafos previos y se tiene una expectativa sólida sobre los resultados a obtenerse.

9.2. Análisis de la viabilidad del proyecto de fin de carrera

Los apartados siguientes detallan los factores de viabilidad considerados.

9.2.1. Viabilidad técnica

Primero, existen los recursos informáticos en la Pontificia Universidad Católica del Perú para poder implementar la solución expuesta en el presente trabajo. En detalle, en el 2015 la universidad adquirió una computadora capaz de utilizar una GPU NVIDIA GRID de arquitectura Kepler (última generación en su tipo) que servirá para el entrenamiento y desarrollo de la red neuronal a implementar. Para utilizarlo, los accesos requeridos ya han sido aprobados y probados. Además, la Universidad también cuenta con el software necesario, detallado en el apartado de herramientas, instalado en las computadoras de los laboratorios de informática.

Segundo, los conocimientos necesarios para lograr los objetivos han sido dictados a lo largo de la carrera de ingeniería informática: cursos en las ramas de ciencias de la computación, ingeniería de software, investigación de operaciones, entre otros que hacen factible el desarrollo de la solución propuesta. Por otro lado, el autor de esta tesis trabaja actualmente en finanzas corporativas, con lo cual se cuentan con los conocimientos en el área financiera, y también con la experiencia y los contactos adecuados para resolver dudas puntuales.

Por último, el asesor del presente trabajo cuenta con una larga carrera de investigación y estudios en el área de aprendizaje de máquina que ayudarán al desarrollo de la solución ideada.

Tabla 3: Viabilidad económica

Tarea	Costo HH
Total Proyecto	439
1. Fase de concepción	109
1.1. Problemática	16
Definición del problema	16
Investigación del estado de arte	0
1.2 Marco Conceptual y Objetivos	0
Desarrollo del marco conceptual	0
Definición de objetivo general y específicos	0
1.3 Resultados y herramientas	31
Definición de resultados esperados	7
Investigación dsobre herramientas	24
1.4 Alcance, limitaciones y viabilidad	62
Definición de alcance y limitaciones	28
Análisis de riesgos	30
Desarrollo de la justificación del proyecto	0
Análisis de viabilidad	4
2. Fase de desarrollo	330
2.1 Diseño	70
Diseño de la arquitectura de red neuronal	40
Diseño de los tipos de red neuronal	30
2.2 Extracción de datos	80
Adaptación del módulo de integración	60
Estructuración de la información	20
2.4 Implementación	150
Implementación de la red neuronal extendida propuesta como solución	126
Entrenamiento de las redes neuroanles	20
Consolidación de resultados	4
2.5 Análisis comparativo	30
Análisis de precisión	10
Desarrollo del análisis de rentabilidad	20

En conclusión, el presente proyecto es viable dentro del cronograma expuesto y dentro del rango de horas esperadas para un trabajo de fin de carrera. La métrica principal es el costo HH medidas durante el transcurso del proyecto.

En resumen, se cuenta con los conocimientos, asesoría y experiencia requeridos para el logro de los objetivos propuestos. Así mismo, no se necesita presupuesto y las herramientas están disponibles en la universidad o son gratuitas; es decir, se cuenta con los recursos necesarios. Por último, las fechas definidas en el cronograma están en línea con los objetivos y expectativas propuestos. En conclusión, el proyecto es completamente viable.



10. Extracción y estructuración de información

En este capítulo se revisa el proceso de obtención de datos desde diferentes portales web, los atributos obtenidos y la estructuración de esta información como parte de la preparación para las siguientes fases. El objetivo de este capítulo es la obtención de eventos estructurados a partir de los títulos informativos de las noticias publicadas en portales en línea. Este capítulo está relacionado con el objetivo específico #2.

10.1. Descripción

La fase de extracción y estructuración de información es el punto de inicio en todo diseño y evaluación de modelos neuronales, en esta línea, es vital tener datos e información altamente fiable y de calidad pues de esto dependerá en gran medida el éxito o fracaso del entrenamiento y comparación de las redes neuronales. Esta fase, se inicia recolectando información –títulos de noticias, precios de valores y fechas- tal cuál se encuentra en los portales en línea y se sigue con la estructuración para poder finalmente obtener información del tipo ((Sujeto, Predicado, Objeto); Fecha; Precio; Movimiento) – Ver Imagen 3.

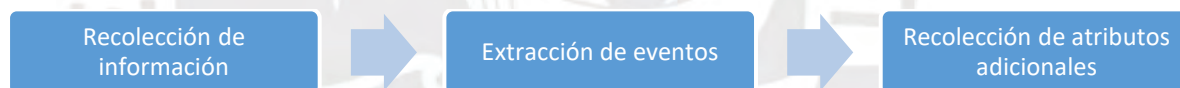


Imagen 3: Flujo de la extracción y estructuración de información

10.1.1. Recolección de información

La información ha sido recolectada mediante una araña web escrita con ayuda de la librería Scrapy en Python (ver Pseudocódigo 1). Los objetivos de esta araña son los siguientes:

- ✓ Recopilar la información disponible en diferentes portales web de noticias.
- ✓ Recopilar nueva información que eventualmente será mostrada en los portales web de noticias.

La información obtenida por esta araña tiene la siguiente estructura:

Título de la noticia, sello de tiempo → “P&G sues IRS”, 05/04/2017

Esta información fue guardada en un archivo CSV para su procesamiento en las siguientes fases del presente trabajo. Cabe mencionar que los títulos de las noticias fueron extraídos de

Google News, Bloomberg y Reuters, y se obtuvo un total de poco más de diez millones de títulos de noticias en una semana ejecutando tres arañas –una por cada portal–.

Pseudocódigo 1: Araña web

```
Inicio araña_recopilar()
    url="www.reuters.com"
    dominio="www.reuters.com"

    Para cada NOTICIA en respuesta_http(url)
        titulo=NOTICIA.Seleccionar("titulo")
        fecha=NOTICIA.Seleccionar("fecha")
        url=NOTICIA.Seleccionar("Siguiente url")
    fPara
faraña_recopilar
```

10.1.2. Extracción de eventos

Esta subetapa se realiza con la ayuda de la herramienta de extracción de información llamada Stanford Open IE. Esta es una aplicación de código abierto de la Universidad de Stanford que permite la estructuración de oraciones a través de la implementación del sistema de extracción de información novedoso basado en un análisis sintáctico (ver Imagen 4) y un clasificador de regresión logística multinomial que obtiene mejores resultados que la literatura y trabajos actuales en este campo (mejores que el modelo de Ollie, el mejor modelo antes de este) [GABOR, 2015].

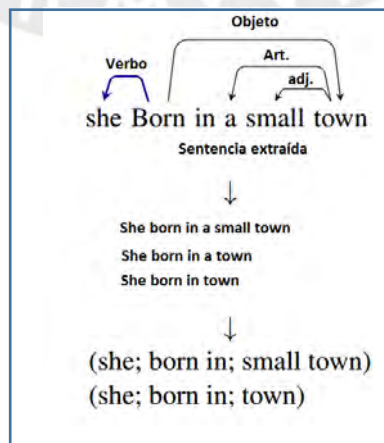


Imagen 4: Composición semántica [GABOR, 2015]

Entrada: P&G sued IRS.

Salida: (P&G; sues; IRS).

(Sujeto; Acción; Objeto)

De esta manera, se estandariza la extracción de información eliminando palabras como preposiciones, conectores, etc. Además, se estructura en términos de sujeto, predicado y objeto, que son relaciones más cercanas a cómo se razona frente a este tipo de información. Cabe mencionar que trabajos anteriores (relatados en el apartado del estado del arte) trataban de predecir el movimiento de la bolsa considerando solo las palabras y su representación vectorial, sin tomar en cuenta las relaciones que existen entre ellas.

Estas relaciones servirán como parte de la entrada de la red neuronal (después de ser representadas en vectores). La otra parte será el tiempo en que sucede este evento para relacionarlas con el movimiento de la bolsa bursátil.

(Sujeto; Acción; Objeto; Tiempo)

(Google; presenta; gadget; 01/01/2001 06:00:00)

10.1.3. Obtención de parámetros adicionales

Adicionalmente a los eventos ya extraídos, se requiere información sobre la tendencia y estado actual/movimiento de las acciones a evaluar, la cual servirá como entrada para las redes neuronales a considerar. Para descargar esta información, se requirió la ayuda de la Interfaz de Programación de Aplicaciones (API) de Yahoo Finance, que permite descargar y trabajar con información financiera en tiempo real sobre las acciones y diferentes mercados bursátiles [YAHOO API, 2016]. De esta manera, se descarga el precio de una acción (incluyendo su símbolo) y el tiempo que estará disponible a ese precio.

“PG”, 82.3, +1.3%, 05/04/2017

11. Representación vectorial de eventos

La representación vectorial es clave para el desempeño del modelo neuronal. Esta debe representar todas las características que se quieren tomar en cuenta. El presente trabajo toma como prioridad las características de cercanía de palabras parecidas (por ejemplo, el vector de “perro” y “can” deben estar cerca), lejanía de palabras opuestas (por ejemplo, el vector de “perro” y “carro” deben estar distantes), relación entre palabras (por ejemplo, el vector de “China” y “Pekín” deben estar a una distancia similar de “Italia” y “Roma”) y densidad (como vectores con pocos ceros en vez de muchos) (Ver Imagen 5). Este apartado está relacionado con el Objetivo #2.

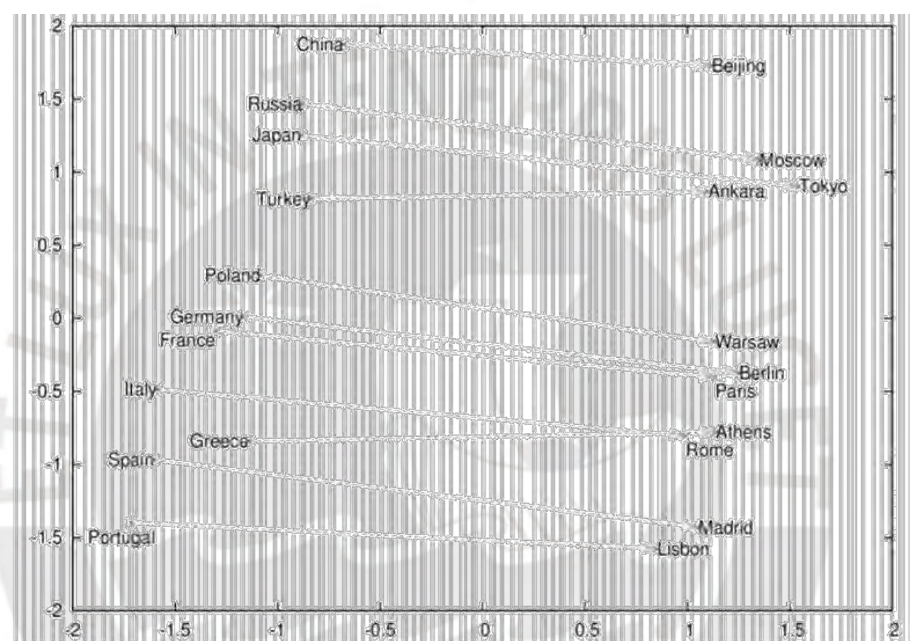


Imagen 5: Representación de palabras [Word2vec, 2017]

11.1. Descripción

En esta fase, el objetivo es encontrar una representación vectorial para cada relación de Sujeto→Predicado→Objeto, de manera que represente adecuadamente las características explicadas en la introducción. Para lograr este objetivo, esta fase consiste en dos actividades. Primero, se representan vectorialmente las palabras individuales mediante el algoritmo de *Skip-Gram* con ejemplificación negativa, propuesta por Mikolov, tomando un modelo preentrenado por Google. Segundo, se representan las relaciones de estas palabras mediante los eventos estructurados obtenidos en el capítulo anterior y una red neuronal tensorial [MIKOLOV, 2013].

11.2. Representación vectorial de palabras

Para la representación vectorial de las palabras, se usa la arquitectura de red neuronal llamada Skip-Gram con ejemplificación negativa, implementada dentro de los modelos Word2Vec. La arquitectura Skip-Gram representa cada palabra en un vector de acuerdo a su contexto. De esta manera, genera vectores similares para palabras similares a través de sus contextos. Por ejemplo, la palabra “inteligente” tendrá un vector similar al de “brillante”, puesto que ambos términos pueden ser usados en contextos similares. Sin embargo, este vector será más distante del que representa a la palabra “avión”. Esta vectorización de palabras se logra mediante el entrenamiento de la arquitectura Skip-Gram, que consiste en una red neuronal de dos capas, que permite la vectorización de palabras individuales de acuerdo al contexto en que van acompañadas. Es decir, cuando una palabra es ingresada en la red neuronales el modelo se entrena utilizando las palabras que están en su contexto para encender las neuronales correspondientes en la capa de salida (Esta metodología trata de imitar como una persona aprende palabras desde un diccionario: definiéndola mediante otras) [MIKOLOV, 2013]. (Ver Imagen 6).

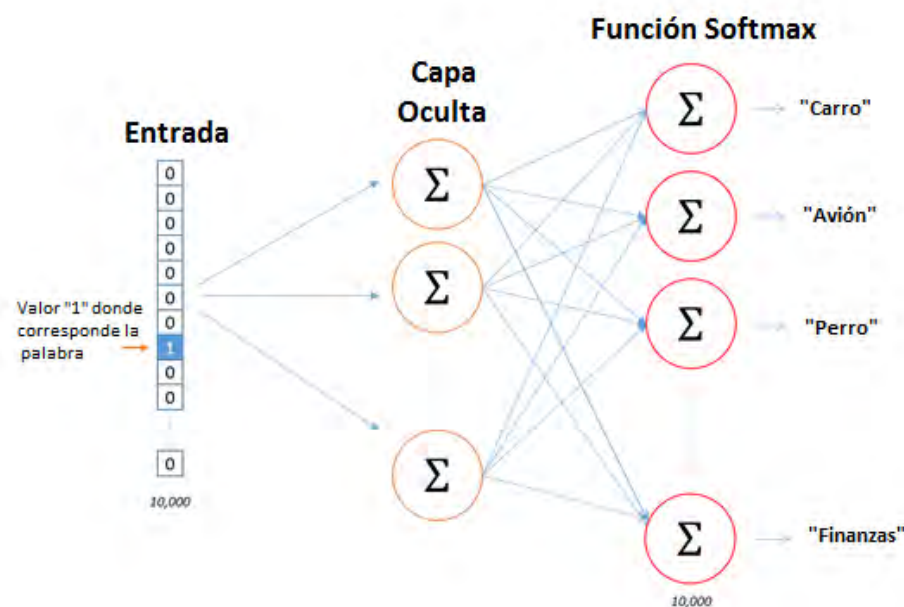


Imagen 6: Skip-gram [McCorminck, 2016]

Existen dos aclaraciones con respecto al uso de Skip-Gram en el presente trabajo. Primero, se utiliza la versión de ejemplificación negativa para optimizar el entrenamiento de esta primera red neuronal. Esta mejora consiste en representar palabras compuestas mediante un solo vector (por ejemplo, “Barack Obama” será un solo vector en vez de dos) y seleccionar aleatoriamente cinco palabras que no están en el contexto para actualizar sus pesos con 0 en vez de calcular el gradiente mínimo (con el fin de optimizar el proceso de entrenamiento). Segundo, se utilizan los pesos base pre entrenados de Google bajo el nombre de Modelo de

Noticias de Google – *Skip-gram (Google News Model Skip-gram)*, el cual contiene un vocabulario de tres millones de palabras entrenadas en el idioma inglés con un conjunto de datos de cien mil millones de palabras publicadas en diferentes artículos y noticias [MIKOLOV, 2013] [WORD2VEC, 2017].

Por último, la red neuronal ha sido ajustada nuevamente con los eventos de noticias recogidas de portales web. De esta manera, se obtiene una red neuronal mucho más precisa y optimizada para el ámbito donde se va a trabajar (predicción de movimiento bursátil).

11.3. Representación vectorial de relaciones

En este apartado se detalla el proceso de transformación de la representación vectorial de palabras individuales (obtenidas en el paso previo) a vectores que representen cada relación de estas palabras obtenidas mediante la extracción y estructuración de los títulos de noticias. Así, para lograr este objetivo se utiliza una modificación de un tipo de red neuronal muy novedoso, original y reciente llamado red neuronal tensorial (RNT), que fue presentado por primera vez por los investigadores Socher R. y Chen D. para completar bases de conocimientos [SOCHER, 2013].

La RNT utiliza tensores en vez de matrices como soporte para los pesos con el fin de obtener más características y patrones de las relaciones. Este tipo de red neuronal obtiene los mejores resultados para la representación vectorial de relaciones, puesto que es un modelo general de trabajos destacados anteriores. El trabajo de Socher y Chen presenta un tensor para relacionar el sujeto, predicado y objeto de las relaciones, mientras que el presente trabajo propone utilizar cinco tensores para representar mejor la relación de la siguiente manera ((Sujeto, Predicado, Objeto); (Sujeto, Predicado); (Predicado, Objeto)) (Ver imágenes 7 y 8).

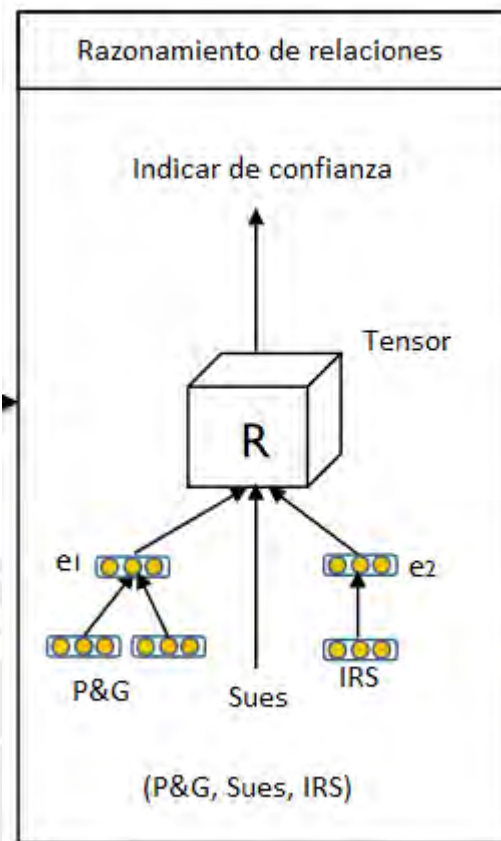


Imagen 7: Red neuronal tensorial [SOCHER, 2013]

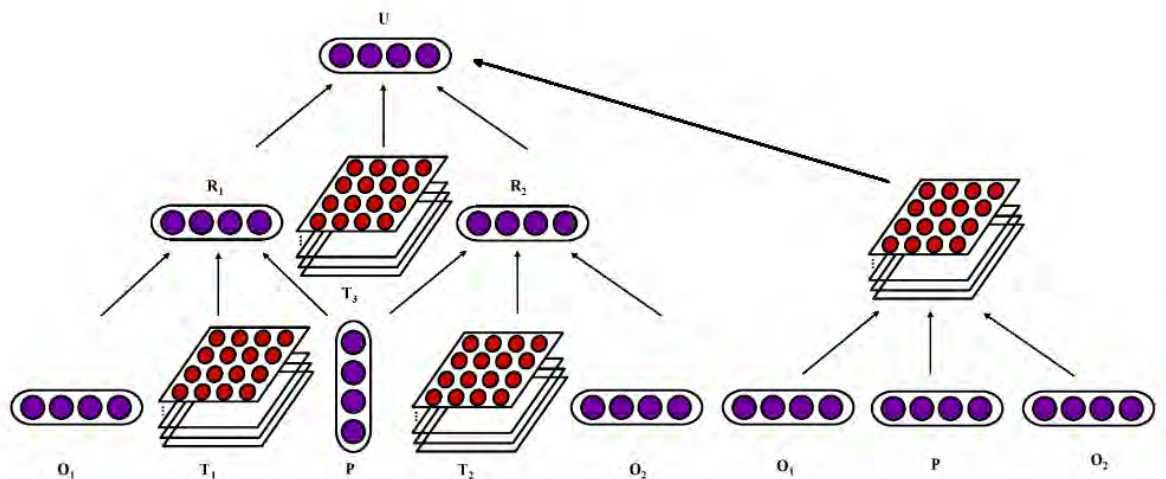


Imagen 8: Red neuronal tensorial adaptada [DING, 2015]

Las entradas (O₁, P, O₂) de esta red neural propuesta son los vectores calculados en el paso previo mediante el algoritmo *Skip-gram*. Las variables R₁ y R₂ son vectores temporales que representan el cálculo hecho en los tensores de la primera capa. El vector U representa el vector final que representa el evento entrante. Un punto clave que se detalla en este paso es el entrenamiento de este tipo de red neuronal (propuesto por Socher y Chen).

Para el entrenamiento de esta red neuronal se han creado conjuntos de datos corruptos con el fin de contrastarlos con los conjuntos correctos, así, si la diferencia de vectores entre un

conjunto corrupto y un conjunto correcto es mayor que 1 se actualizan los pesos de los tensores Ver Pseudocódigo 2.

$$f(E) = f(e_1, R, e_2) = \mu_R^T \tanh \left(e_1^T W_R^{[1:k]} e_2 + V_R \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + b_R \right)$$

Ecuación 2: Función de evaluación de la red tensorial

- ✓ e1: Vector 1 → Sujeto
- ✓ R: Vector 2 → Predicado
- ✓ e2: Vector 3 → Objeto
- ✓ W: Tensor
- ✓ V: Vector de pesos entre e1 y e2
- ✓ bR: Vector de bias

Pseudocódigo 2: Entrenamiento de la red tensorial propuesta [DING, 2014]

```

Entrada: C = (E1, E2, E3, ..., En) un conjunto de eventos obtenidos
Salida: Modelo entrenado RTN'
Crear conjunto Cc = (E1c, E2c, E3c, ..., Enc) corrupto reemplazando algún término de cada
evento aleatoriamente
Mientras C ≠ ∅ hacer
    Pérdida ← max(0, 1 - f(Ei) + f(Er)) + λ || Φ ||22
    Si pérdida > 0 entonces
        Actualizar Pesos
    Sino
        C ← C / {Ei}
Retornar RTN'
  
```

12. Modelo de red neuronal recurrente profunda con memoria Largo-Corto plazo

Actualmente, se han desarrollado modelos recurrentes de redes neuronales profundos con memoria de largo-corto plazo que han dado resultados brillantes, los cuales van desde el reconocimiento de voz, síntesis de voz humana, predicción de estructura proteica hasta, en los videojuegos, un nivel superior al de un jugador experto [MNIH, 2013] [GOOGLE, 2015]. En esta línea, este apartado describe la fase central del presente trabajo, en donde se procesan los eventos representados mediante vectores como entradas para una red recurrente profunda con memoria de largo-corto plazo para obtener la predicción de subida o bajada en una bolsa de valores específica. Este apartado apunta a los objetivos #1, #3 y #4.

12.1. Arquitectura de red neuronal

La arquitectura propuesta para aprender mediante los vectores de eventos obtenidos en el apartado anterior es una red recurrente que usará unidades de memoria de largo-corto plazo (ver Imagen 10). Este modelo está representado en la imagen 9. Las características y configuración de la presente arquitectura están representadas en la tabla 4.

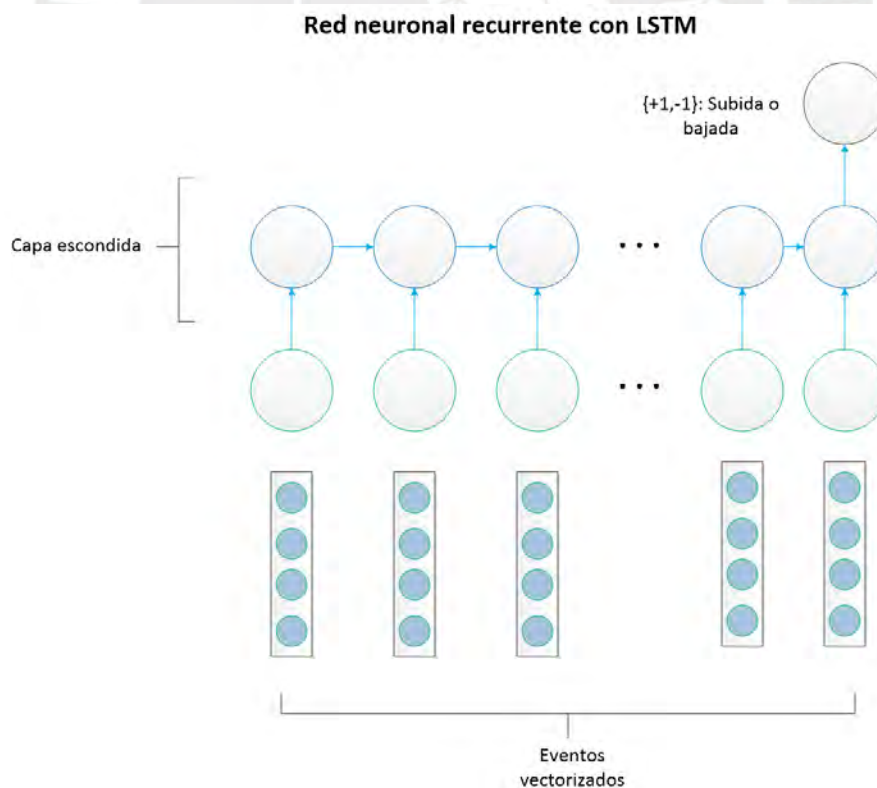


Imagen 9: Arquitectura de red neuronal recurrente

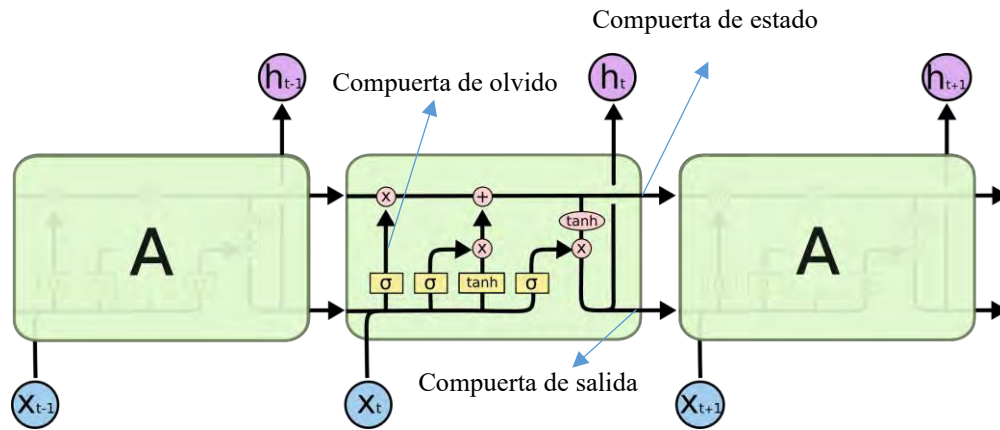


Imagen 10: Unidad LSTM, X como entrada y h como salida [COLAH, 2015]

Tabla 4: Configuración de la arquitectura

Parte del modelo neuronal	Cantidad	Tipo de unidad
Capa de entrada	365 unidades de entrada	Vectores de eventos de 300 características
Capas escondidas	1 capa escondida	
Capa de salida	1 Unidad de Salida	1 si la tendencia es positiva o -1 si la tendencia es negativa
Tipo de unidad	-	LSTM

En detalle, y como justificación, esta configuración de la arquitectura propuesta toma como entrada 360 unidades para representar la influencia de los eventos a lo largo del tiempo. Así, cada unidad representa el vector promedio de los eventos representados mediante vectores de un solo día, siendo los primeros los más antiguos (a la izquierda en la imagen 9) y los últimos, los más recientes (a la derecha en la imagen 9). De este modo, no solo se representa la dependencia temporal, puesto que las unidades con eventos más recientes estarán afectadas también por los eventos más antiguos, sino que también hace que la granularidad de espacios temporales sea mucho mayor (en vez de considerar tres unidades: una para largo, otra para mediano y la última para corto plazo) (ver Imagen 10.). Por otro lado, se ha escogido el tipo de unidades de memoria de largo-corto plazo / redes recurrentes profundas no solo por sus impactantes resultados actuales en diferentes áreas [KARPATHY, 2015], sino también, por las características de manejo del tiempo, volatilidad, precisión y manejo de secuencias de tiempo que están presentes en los movimientos bursátiles [GRAVES, 2008].

De este modo, es necesario definir las compuertas usadas (ver Imagen 10 como guía):

- Compuerta de olvido: Está compuerta está diseñado con el objetivo de clasificar que información será olvidada y que información será mantenida. Para ello se hace

uso de la función sigmoide (valor de salida entre 1 y 0) como muestra la siguiente Ecuación:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f)$$

Ecuación 3: Compuerta de olvido [COLAH, 2015]

- ✓ σ Función Sigmoidal
- ✓ W Pesos de la compuerta
- ✓ h_{t-1} Salida de la celda en el paso temporal anterior
- ✓ X_t Entrada de la celda en el paso temporal actual
- ✓ b Bías de la compuerta

- Compuerta de estado de célula: Está compuerta está diseñada con el objetivo de decidir qué información se guardará como estado de la célula LSTM, como se presenta en la siguiente Ecuación la función de cálculo de este estado está basado no solo en la entrada y estado de la anterior célula, sino también, del resultado de la compuerta de olvido (f_t):

$$C_t = f_t * C_{t-1} + \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

Ecuación 4: Compuerta de estado de célula [COLAH, 2015]

- ✓ σ Función Sigmoidal
- ✓ W Pesos de la compuerta
- ✓ h_{t-1} Salida de la celda en el paso temporal anterior
- ✓ X_t Entrada de la celda en el paso temporal actual
- ✓ b Bías de la compuerta
- ✓ Tanh Función tangente hiperbólica

- Compuerta de salida: Con esta compuerta se calcula finalmente cuál será la salida de la celda, la salida depende de la entrada, el estado de la célula anterior y el estado de la celda actual como se muestra en la siguiente Ecuación:

$$h_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) * \tanh(C_t)$$

Ecuación 5: Compuerta de salida

- ✓ σ Función Sigmoidal
- ✓ W Pesos de la compuerta
- ✓ h_{t-1} Salida de la celda en el paso temporal anterior

- ✓ X_t Entrada de la celda en el paso temporal actual
- ✓ b Bias de la compuerta
- ✓ Tanh Función tangente hiperbólica

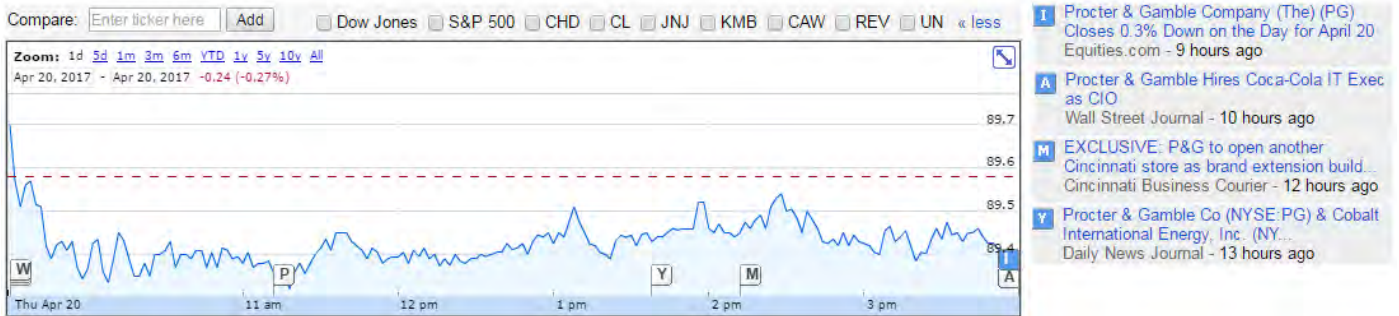


Imagen 11: Efecto de las noticias en la bolsa bursátil [GOOGLE, 2017]

12.2. Implementación e integración del módulo

Como se mencionó en el apartado anterior, la entrada a este modelo neuronal son los eventos representados como vectores de 300 características, obtenidos mediante la red tensorial del capítulo anterior. Estos vectores han sido guardados en un archivo llamado CSV_de_eventos.csv, que se muestra en la tabla 4. Por otro lado, las etiquetas de cada salida, que servirán para el entrenamiento de la red neuronal, se leerán desde otro CSV mostrado como CSV_de_movimientos.csv. Como ya se mencionó, si bien el presente trabajo podría ser ejecutado para acciones particulares con una entrada de noticias enfocadas en corporaciones como Walmart, P&G, Colgate, etc., se ejecuta la experimentación solo para el índice S&P. Por lo tanto, se consideran las noticias en general para ver el impacto en el movimiento general de la bolsa bursátil. Así mismo, las etiquetas como resultado también son valores y movimientos del índice S&P. El manejo de estos archivos CSV se realiza, como en los apartados anteriores, mediante la librería OpenCSV. Por último, el modelo se implementa mediante el uso de la librería DeepLearning4J, utilizando como pilares las clases GravesLSTM y BPTT (creadas en base a la red LSTM descrita en la tesis de Alex Graves [GRAVES, 2015]) (ver pseudocódigo 3). Los pesos producto del entrenamiento de estas redes serán almacenadas en el componente Pesos_RNN.csv (Ver tabla 5).

Tabla 5: Descripción de componentes

Componente	Peso	Descripción	Parámetros
CSV_de_eventos_vectorizados.csv	70.4 GB	Contiene la entrada de la red neuronal	Fecha (AAAA/MM/DD EST, Vector evento – 300 características)
CSV_de_movimientos.csv	511 MB	Contiene la salida correcta de la red neuronal	Fecha (AAAA/MM/DD EST, Movimiento del S&P durante el día, máximo, mínimo)
Pesos_RNN.csv	5.1GBs	Contiene los pesos de la red recurrente LSTM entrenada	Pesos por Unidad

Pseudocódigo 3: Implementación de la red neuronal recurrente LSTM

Entrada: $C = (E_1, E_2, E_3, \dots, E_n)$ un conjunto de vectores de eventos por día, Siendo E_1 el más antiguo y E_n el más reciente

$O = (O_1, O_2, O_3, \dots, O_n)$ un conjunto de clases (+1,-1) de movimientos del S&P por día, Siendo O_1 el más antiguo y O_n el más reciente

Salida: Modelo entrenado RNN-LSTM (Red neuronal recurrente con LSTM) y archivo de pesos="pesos_RNN.csv".

Inicializar configuración de RNN-LSTM (3 capas – Imagen 6)

Inicializar pesos aleatorios

Sea $n = \text{Elementos de } C$

Mientras ($i=0$ y $i < n-365$)-hacer

Mientras ($j=1$ y $j < 365$) hacer

$C' = (E_{1+i}, E_{2+i}, E_{3+i}, \dots, E_{i+365})$

Calcular Salida (h) y Estado de salida (SC) (Fórmulas LSTM)

Error = Error + $\{O_i - h\}$

$j \leftarrow j+1$

fMientras

Aplicar BPTT (Error, RNN-LSTM) (Ver pseudocódigo y Ecuación BPTT)

$i \leftarrow i+1$

fMientras

Imprimir MCC y Precisión

Retornar RNN-LSTM'

12.3. Entrenamiento y validación cruzada con K iteraciones

Para llevar a cabo el entrenamiento de la red neuronal se hace uso del algoritmo de entrenamiento llamado programación hacia atrás a través del tiempo, que es una extensión del algoritmo de propagación hacia atrás. En detalle, toda red neuronal puede ser entrenada con propagación hacia atrás siempre y cuando la función de costo sea diferenciable. La más común, y la que el presente trabajo utilizará, se llama suma de errores cuadráticos (Ver Ecuación 6), de esta manera, para la optimización de los pesos de la red neuronal se usa un proporcional de la gradiente negativa de la función de costo (i.e. el factor de proporción es el ratio de aprendizaje. Ver Ecuación 6).

Adicionalmente, esta gradiente no solo depende de la función de costo, sino también, de la función de activación usada, que también tiene que ser diferenciable para poder hallar su derivada en el cálculo de la gradiente [GUO, 2013].

$$C = \frac{1}{2} \sum_p^n \sum_k^o (d_{pk} - y_{pk})^2$$

Ecuación 6: Función de costo “Suma de errores cuadráticos” [GUO, 2013]

- ✓ n número total de ejemplos de entrenamiento
- ✓ k índice de las unidades de salida
- ✓ p índice de las unidades de ejemplo
- ✓ o número de unidades de salida

$$\Delta\omega = -\eta \frac{\partial(C)}{\partial(\omega)}$$

Ecuación 7: Gradiente descendiente para actualizar los pesos de la red neuronal [GUO, 2013]

- ✓ ω parámetro de pesos de la red neuronal
- ✓ η ratio de aprendizaje
- ✓ C función de costo

En esta línea, la extensión del algoritmo BP para redes neuronales recurrentes es el llamado BPTT. Este tipo de propagación hacia atrás se hace en todas las capas anteriores, a diferencia de BP que solo

toma las dos últimas, para capturar la influencia de la influencia histórica. (Red recurrente desplegada en imagen 9). Esta programación hacia atrás a través del tiempo es mostrada en la Ecuación. Adicionalmente, sin la característica de memoria de largo-corto plazo (LSTM) el algoritmo BPTT tendría un efecto mínimo o caótico en las capas más profundas debido al problema “Desaparición de gradiente”.

$$\delta_{pj}(t-1) = \sum_h^m \delta_{pj}(t) * \mu_{hj} * f'(s_{pj}(t-1))$$

Ecuación 8: Programación del error hacia atrás en el tiempo [GUO, 2013]

- ✓ t paso del tiempo
- ✓ h índice del nodo escondido en el tiempo t
- ✓ j índice del nodo escondido en el tiempo $t-1$
- ✓ p índice del ejemplo de entrenamiento
- ✓ s peso del estado del nodo en el tiempo $t-1$
- ✓ u peso de la capa del escondida que entre t y $t-1$

Cómo se detalló en el apartado anterior la red neuronal recurrente profunda consta de 365 nodos de entrada, una capa escondida de 365 nodos también, y por último un nodo de salida (Ver imagen 9), cabe resaltar que cada entrada es un vector de 300 características, resumiendo, el total de pesos a entrenar en la red neuronal es 32'850'000. Para llevar a cabo este entrenamiento, se dividió el archivo de entrada (“CSV_de_eventos_vectorizados.csv”, ver Tabla 4) que consta de 13'347'890 eventos vectorizados (i.e. ejemplos de entrada) de manera que el 70% (i.e. 9'343'523) es usada para el entrenamiento y el 30% (i.e. 4'004'367) será usada para el test. De esta manera, se realizará la validación cruzada con 5 iteraciones para poder determinar la precisión del pronóstico. Los resultados del entrenamiento, predicción y capacidad para hacer ganancias de la red neuronal se discutirán en el siguiente capítulo.

12.4. Implementación de la simulación en el mercado bursátil

La otra faceta del presente trabajo es la rentabilidad financiera obtenida de este modelo en una simulación de mercado, debido a que la rama en la que se está desarrollando esta tesis es ciencias de la computación, se utilizará una estrategia de simulación de inversión de compra / venta de acciones básica usada en trabajos anteriores de este tipo.

Así, la estrategia de inversión describe que se debe comenzar con un capital ficticio de 10'000 USD siendo los procesos de compra y venta con las siguientes reglas [DING, 2014] [LAVRENKO, 2000].

- ✓ Durante el día de simulación si el inversor sabe que el índice subirá, él compra el índice al precio al que abre y si durante ese día sube un 2% lo vende inmediatamente, sino lo vende al precio de cierre.
- ✓ Durante el día de simulación si el inversor sabe que el índice bajará, él usar el método “*shorting*”, es decir, se prestará el índice al inicio del día y lo venderá al precio al que abre, y si durante el día baja 1% comprará la misma cantidad de acciones, en caso contrario, comprará con el precio de cierre para hacer el encubrimiento.
- ✓ Debido a que las comisiones varían mucho de acuerdo a la cantidad intercambiada y dependiendo del bróker, no se considera el uso de comisiones en la simulación. Para un mayor entendimiento referirse a las imágenes 12 y 13, así mismo, se muestra en el pseudocódigo 4.



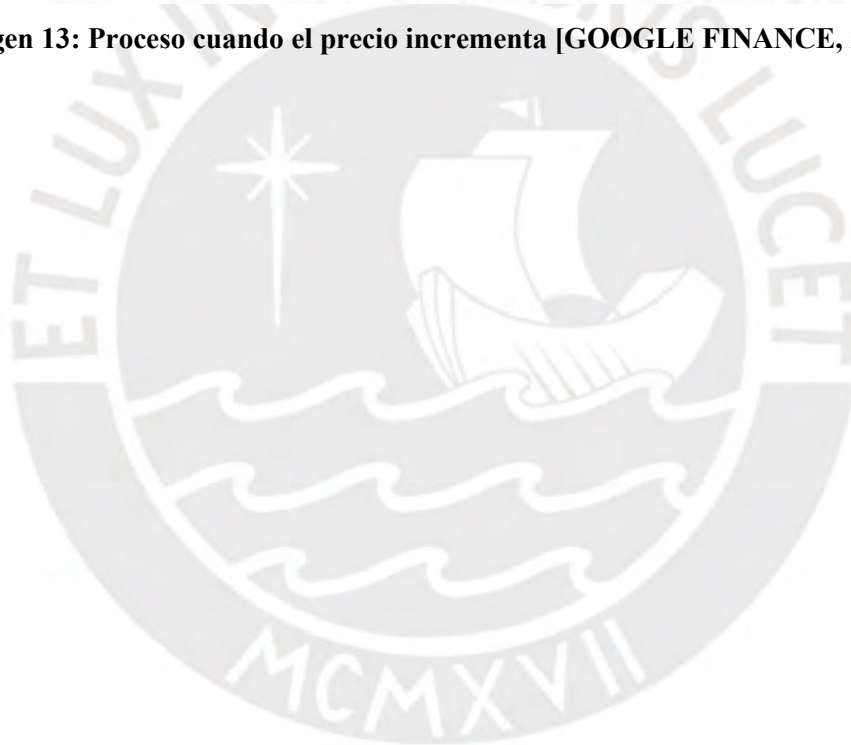
Imagen 12: Proceso de “shorting” – Precio decrece [GOOGLE FINANCE, 2017]

S&P 500

2,348.69 ↓7.15 (0.30%)



Imagen 13: Proceso cuando el precio incrementa [GOOGLE FINANCE, 2017]



Pseudocódigo 4: Implementación de la estrategia de inversión

Entrada: $P = (P_1, P_2, P_3, \dots, P_n)$ un conjunto de clases (+1, -1) que representa la predicción de la red neuronal.

$O = (O_1, O_2, O_3, \dots, O_n)$ un conjunto de clases (+1,-1) de movimientos del S&P por día reales.

Salida: R=Rentabilidad obtenida en la implementación del presente trabajo

R=10 000 //10 000 representa el capital actual de la inversión (se descontará luego)

Para cada día (d) desde 01/01/2007 hasta 31/08/2016 hacer

$A = \text{S\&P 500}(d,1)$ //Apertura del valor del índice al minuto 1

Si $(R \leq 0)$ entonces

Retornar “Quiebra”

fSi

Si $(P_d = +1)$ entonces

Comprar acción al precio de entrada

Sino

Vender acción al precio de entrada

fSi

Para cada minuto (m) del día d hacer

V=Valor del índice S&P 500(d,m)

Si $(P_d = +1)$ entonces

Si $((V/A-1) > 2\%)$ entonces

Vender acción

$R = R * 1.2$

fSi

Sino

Si $((A/V-1) < 1\%)$ entonces

Comprar acción

$R = R * 1.1$

fSi

fSi

fPara

C=Valor del índice al cierre

Si (Pd=+1 && No compra / venta antes) entonces

Vender acción al precio de salida

Sino

Comprar acción al precio de entrada

fSi

$R=R*(C/A-1)$

fSi

fPara

Retornar R-10 000



13. Experimentación y resultados

El presente capítulo detalla las observaciones obtenidas en la experimentación del modelo neuronal completo (combinación conjunta de las redes neuronales expuestas en capítulos anteriores). Se utilizan tres indicadores para analizar el desempeño de la propuesta: medición de precisión mediante la validación cruzada con 5 iteraciones, coeficiente de correlación de Matthews y el promedio de la rentabilidad obtenida en nueve años de experimentación. Apunta al objetivo #4.

13.1. Observaciones y configuración de la experimentación

Como se detalló en los capítulos anteriores, para llevar a cabo la experimentación se dividió el trabajo en tres fases principales.

Primero, se extrajeron y estructuraron las noticias desde Google News y Reuters. Se obtuvo alrededor de diez millones de títulos que fueron estructurados en eventos del tipo (Sujeto, Predicado, Objeto) mediante la investigación publicada bajo el título de “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank” [SOCHER, 2013]. Cabe resaltar que, en esta etapa, hubo un subproceso de limpieza de datos, en el cual se eliminaron los títulos de noticias que estuvieran vacíos. En total se eliminaron 127 registros.

Segundo, una vez obtenidas las noticias estructuradas, se ejecutaron dos procesos. En primer lugar, se obtuvieron los vectores de cada palabra en cada evento estructurado, para lo cual se hizo uso del modelo preentrenado de Word2Vec [MIKOLOV, 2013]. Así, se obtuvieron vectores de 300 características que pueden representar cada palabra individualmente. Adicionalmente, se hizo una segunda limpieza de datos eliminando las palabras que no pudieran ser representadas (e.g. Mi-171, un modelo de helicóptero militar ruso). En segundo lugar, para capturar el efecto de orden de las palabras y estructura semántica, se hizo uso de la red neuronal tensorial presentada por Socher en el trabajo “Reasoning with Neural Tensor Networks for Knowledge Base Completion” [SOCHER, 2013] cuyo objetivo es solucionar el problema de datos incompletos en las bases de conocimientos. De este modo, se captura mediante un solo vector de 300 características la representación de un evento preservando su estructura y orden de palabras.

Por último, estos vectores que representan cada evento estructurado sirven como entrada para el entrenamiento de la red neuronal profunda recurrente con memoria de largo-corto plazo, adaptada de la arquitectura de red neuronal recurrente LSTM general y expuesta en la tesis de Alex Graves [GRAVES, 2008], para la predicción del movimiento del índice

S&P 500 de la bolsa bursátil de New York. Así, se usan 365 vectores de eventos que, como se mostró en el capítulo anterior, representan cada uno un día durante un año.

13.2. Experimentación sobre modelos predictivos

En el capítulo once se obtuvieron los eventos ya representados mediante vectores después de un proceso recolección y limpieza de datos, estructuración de oraciones y representación vectorial de palabras. En este punto, es necesario un modelo para asociar estos eventos representados mediante vectores con los movimientos del índice S&P 500. Por ello, se experimentó con dos tipos de redes neuronales profundas: redes neuronales convolucionales y redes neuronales recurrentes (LSTM), con el fin de escoger entre ellas la que mejor precisión ofrece.

Debido al tiempo que demora el entrenamiento de una red neuronal profunda, para la toma de decisión sobre el modelo neuronal a usar se ha tomado 2 millones de vectores y los parámetros mostrados en la tabla 6.

Tabla 6: Parámetros de los conjuntos de datos

Parámetro	Entrenamiento	Test
#eventos	1'500'000	. 500'000
#títulos	975'031	365'342
#palabras	64'173'834	2'742'152
Intervalo de tiempo	01/2007 – 02/2009	01/2007 – 02/2009
Duración entrenamiento	1 día 3 horas	8 horas 24 min

La arquitectura de la red recurrente se mostró en el capítulo anterior, mientras que la arquitectura de la red neuronal convolucional se muestra en la imagen 14 y es descrita en la tabla 7.

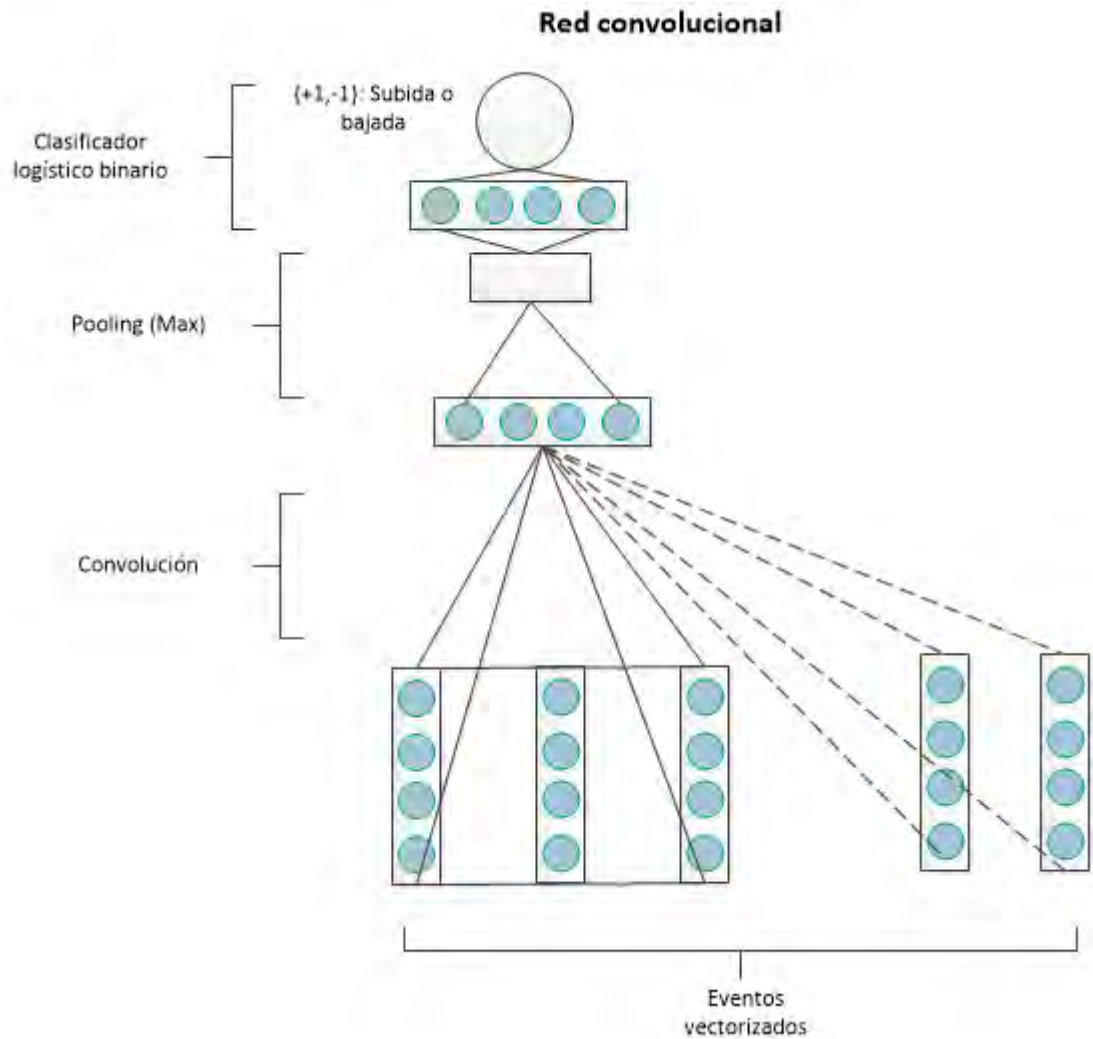


Imagen 14: Red convolucional

Tabla 7: Configuración de la red convolucional

Parte del modelo neuronal	Cantidad	Tipo de unidad
Capa de entrada	365 unidades de entrada	Vectores de eventos de 300 características
Capas escondidas	1 capa de convolución sobre 3 días	
Capa de pooling	Reduce a 3 características con la metodología de máximo	
Capa de clasificación	Clasificador binario logístico	
Capa de salida	1 Unidad de Salida	1 si la tendencia es positiva o -1 si la tendencia es negativa

De esta manera, los resultados del entrenamiento en las medidas de precisión y el coeficiente de correlación de Matthews (MCC) mostradas en la tabla 8 muestran que la

arquitectura recurrente de red neuronal (LSTM) funciona mejor como solución a la problemática planteada en este trabajo. En el trabajo de Ding, se muestra la superioridad de la arquitectura convolucional frente a trabajos previos, estableciendo así una nueva marca en el estado del arte [DING, 2015]. En conclusión, nuestros resultados en este experimento en conjunto con lo expuesto en el trabajo de Ding llevan a tomar la decisión de usar el modelo de red recurrente como parte final del modelo global para la predicción de movimiento del índice S&P 500.

Tabla 8: Resultados MCC y Precisión sobre los modelos convolucional y recurrente

Año	Convolutacional		Recurrente	
	MCC	Precisión	MCC	Precisión
2008	0.38	57.50%	0.36	61.66%
2009	0.46	70.28%	0.44	75.36%
Promedio	0.42	63.89%	0.4	68.51%

13.3. Experimentación del modelo global

Como se detalló en apartados anteriores, la experimentación se basará en dos facetas del presente trabajo: la faceta técnica (precisión de la predicción y el coeficiente de correlación de Matthews) y la faceta financiera (rentabilidad obtenida en la simulación del mercado bursátil de New York).

13.3.1. Experimentación de la predicción en el índice S&P 500

Para obtener los resultados del poder de predicción del índice de S&P 500, se tomó el archivo CSV_de_eventos_vectorizados.csv, obtenido como resultado del flujo a través de las tres primeras redes neuronales (estructuración de eventos, representación vectorial de palabras y la representación vectorial de eventos) y, como etiquetas de cada predicción, el archivo CSV_de_movimientos.csv, obtenido directamente de Yahoo Finance. La experimentación se hizo mediante la validación cruzada de 5 iteraciones sobre cada conjunto de datos (cada conjunto de datos tiene los eventos vectorizados y movimientos bursátiles de un año). En el año 2016, la experimentación fue desarrollada hasta agosto debido a que se fijó ese rango en el conjunto de datos obtenidos (ver Tabla 9).

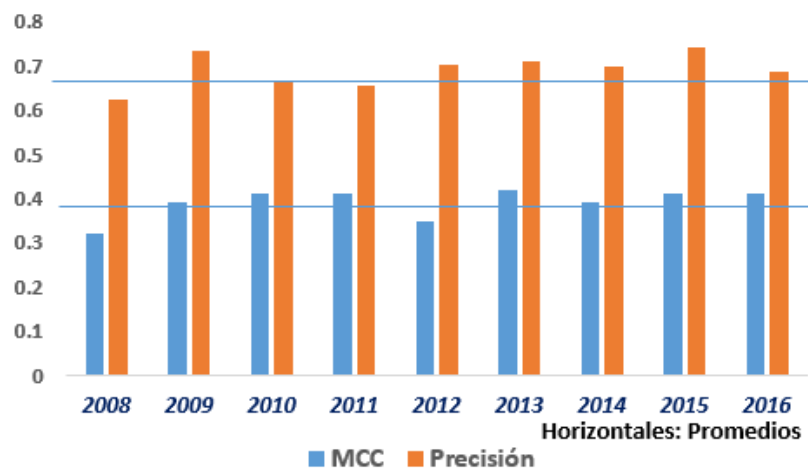
Los resultados fueron obtenidos en un rango de 9 años (2008-2016) mostrados en la tabla 10 y en la imagen 15.

Tabla 9: Parámetros de los conjuntos de datos

	Entrenamiento	Test
#eventos	9'343'523	.4'004'367
#títulos	5'986'007	2'565'433
#palabras	408'702'049	16'675'314
Intervalo de tiempo	01/2007 – 08/2016	01/2007 – 08/2016
Duración	7 días	3 días

Tabla 10: Resultados de la experimentación en las medidas de MCC y precisión.

Año	MCC	Precisión
2008	0.32	62.44%
2009	0.39	73.44%
2010	0.41	66.25%
2011	0.41	65.55%
2012	0.35	70.20%
2013	0.42	71.06%
2014	0.39	69.80%
2015	0.41	74.21%
2016	0.41	68.52%
Promedio	0.39	69.05%

**Imagen 15: Precisión y MCC**

Sobre estos resultados podemos concluir que la precisión de la predicción ronda el 69 % constantemente; sin embargo, decrece en años en los que la bolsa fue impactada por una crisis (2008, 2010 y 2011). Así mismo, la experimentación muestra un coeficiente de correlación de Matthews bastante sólido para el porcentaje de precisión obtenido. Esto demuestra que el presente trabajo mejora los resultados presentados en el estado del arte, lo que sucede por dos razones. La primera es que se utilizó representación vectorial de eventos en vez de usar solo de palabras, ya que la red neuronal toma en cuenta las relaciones semánticas y el orden de las palabras. La segunda es que se utilizaron las redes recurrentes profundas para el pronóstico final por su capacidad para la evaluación del presente tomando en cuenta series de tiempo históricas. Adicionalmente, se usaron redes neuronales novedosas y con los mejores resultados en la literatura para poder estructurar las oraciones en sujeto, predicado y objeto, así como la representación inicial de las palabras en vectores mediante el modelo de Word2Vec, el cual toma en cuenta el contexto en el que se encuentran las palabras para su representación. Así, estas dos mejoras se ajustan mejor al propósito del presente trabajo.

13.3.2. Experimentación mediante la simulación de mercado

En el presente trabajo se simuló la compra y venta del índice S&P 500 mediante la estrategia de inversión básica (descrita en el capítulo anterior) que modela el comportamiento diario de un *trader* [LAVRENKO, 2000]. El capital ficticio inicial para la simulación fue de 10 000 USD. Las ganancias obtenidas en cada año se muestran en la Tabla 11.

Como se puede observar en la tabla, se obtiene una rentabilidad positiva en todos los casos. Sin embargo, al igual que la precisión, esta rentabilidad decrece en los años de crisis (2008 y 2011) debido a que el poder de predicción disminuye en esos años. Es necesario resaltar que esta rentabilidad obtenida no considera las comisiones que están sujetas a la compra y venta de acciones o índices.

Tabla 11: Rentabilidad por año

Año	Rentabilidad
2008	15,669.4
2009	18,237.9
2010	18,011.0
2011	14,907.7
2012	18,350.8
2013	15,417.8
2014	20,253.4
2015	17,265.2
2016	19,702.8
Promedio	17,535.1



14. Discusión

En este capítulo se comparan las extensiones, mejoras e innovaciones que se hacen en este trabajo en comparación a la literatura existente.

14.1. Sobre la estructuración y estructuración de información

Como primer punto de discusión, este trabajo usa una cantidad de noticias superior a las usadas en trabajos anteriores en la literatura. Así, mientras que en el trabajo de Ronny Luss y Alexandre Aspremont llamado “Predicting Abnormal Returns From News Using Text Classification” se consideran solamente 500 artículos en el período del 2000 al 2007 y en el trabajo del equipo del investigador Ding X. llamado “Deep Learning for Event-Driven Stock Prediction” se usan aproximadamente 600'000 noticias, el presente trabajo contempla un total de 8'551'440 noticias para el entrenamiento y validación de la arquitectura de red neuronal profunda propuesta [LUSS, 2012] [DING, 2015]. En consecuencia, el presente trabajo no solo está basado en una base de información mucho más vasta, sino también demuestra que tiene la capacidad para manejar información masiva que guardada en texto plano ocupa más de 70 GB.

Segundo, trabajos pioneros que conforman la literatura actual en esta área no capturan la estructura de las sentencias, por el contrario, el presente trabajo considera lo último de la literatura en la estructuración de oraciones como parte de la estructuración de la información en inspiración a como las personas procesan una frase (Sujeto, Predicado, Objeto) mediante la tecnología de extracción de información Open IE Stanford [GABOR, 2015] [CHEN, 2009] [LUSS, 2012]. Solo el trabajo de Ding X. hace una estructuración de oraciones, sin embargo, no utiliza la última herramienta de extracción de información del estado del arte, probablemente, por haber sido desarrolladas paralelamente [DING, 2015].

Por último, el presente trabajo comienza desde la extracción de información mediante la creación de una araña web, mientras que trabajos de la literatura comienzan con un conjunto de datos sin detallar la obtención o recolectados de manera manual [LUSS, 2012] [KOGAN, 2009] [DING, 2015]. Así, en el capítulo diez se implementó una araña web que puede ser fácilmente adaptada para la recolección de noticias en distintos portales web.

14.2. Sobre la representación vectorial de palabras y eventos

En primer lugar, trabajos de la literatura usan modelos simples para la representación vectorial de palabras, como bolsa de palabras (*bag-of-words*), reconocimiento de entidades nombradas o extracción de frases [KOGAN, 2009] [CHEN, 2009], sin embargo, estos modelos ignoran el contexto de cada palabra lo que descarta información muy importante sobre la semántica de la misma. El trabajo de Ding X. usa el modelo de Skip-Gram desarrollado por Mikolov el cuál es lo último del estado del arte, sin embargo, no solo utiliza 100 características para representar cada palabra en un vector [DING, 2015][MIKOLOV, 2013]. En contraste, este trabajo usa 300 características mediante en línea con la recomendación de los desarrolladores del modelo Skip-Gram mediante la implementación de Google News [WORD2VEC, 2017]. Así, no solo se considera el contexto como parte de la semántica de la palabra, sino también se captura más patrones la considerar un número óptimo de características.

Segundo, el trabajo de estado de arte de Socher R. “Reasoning With Neural Tensor Networks for Knowledge Base Completion” introduce la red neuronal tensorial para la representación vectorial como generalización de modelos previos en la literatura, para demostrar que esta arquitectura obtiene resultados espectaculares que sigue siendo lo último para la representación vectorial de eventos [SOCHER, 2013]. Así, este trabajo sirve como base para la representación vectorial que usa el presente proyecto para capturar el orden, el contexto y la semántica de los títulos de noticias que sirven como entrada, esta innovación solo es considerada en el trabajo de Ding X. donde se utiliza una adaptación del trabajo de Socher R. para la representación vectorial [DING, 2015].

14.3. Sobre el modelo de red recurrente profundo con memoria de Largo-Corto plazo

Como primer punto a discutir, la naturaleza de aprendizaje profundo en procesamiento de lenguaje natural para la predicción de movimiento bursátil del presente trabajo solo ha sido contemplado en el trabajo de Ding X. [DING, 2015], otros trabajos previos solo han considerado modelos más sencillos como máquinas de soporte de vectores [LUSS, 2012] y clasificadores de Bayes [LAVRENKO, 2000]. Y trabajos pioneros han sido basados en tópicos de estadística como el notable trabajo del equipo de LeBaron B. donde utilizan análisis de series de tiempos para predecir la volatilidad del mercado bursátil. En resumen, el presente trabajo utiliza modelos de aprendizaje profundo con información masiva para su entrenamiento, así usa e implementa lo último en el aprendizaje de máquina para empujar la predicción del estado de arte.

Como segundo punto de discusión, dentro del aprendizaje profundo, las redes convolucionales y redes recurrentes con memoria de largo-corto plazo (LSTM) han

demostrado resultados muy importante y nunca antes visto en el reconocimiento de imágenes [LECUN, 1998], aprendizaje de videojuegos [MNIH, 2013], reconocimiento de voz [HINTON, 2012], sintetización de voz [WAVENET, 2016], etc. Así, en la problemática de la predicción del movimiento de la bolsa bursátil mediante modelo de aprendizaje profundo solo el trabajo de Ding X. ha usado redes convolucionales. En contraste, el trabajo presente utiliza por primera vez una arquitectura de red neuronal profunda de memoria de largo-corto plazo (LSTM) con 365 entradas vectoriales para predecir el movimiento del índice S&P 500. Ello ha permitido obtener mejores resultado en la predicción y obtención de rentabilidad, esto debido a que las redes recurrentes son mejores para la predicción con data histórica en vez de las redes convolucionales, las cuales tienen mayor impacto en el reconocimiento de imágenes [GRAVES, 2008].



15. Conclusiones y trabajos futuros

15.1. Conclusiones

Después del desarrollo de la experimentación, se ha podido demostrar que el uso del aprendizaje profundo, en particular usando redes recurrentes LSTM y representación vectorial de eventos, sirve para la predicción de movimientos en los precios bursátiles del índice S&P 500. De este modo, los resultados experimentales de este proyecto presentan que (i) la captura de semántica de eventos (red neuronal tensorial), (ii) la consideración del contexto de las palabras (Word2Vec), (iii) la estructuración semántica y (iv) la consideración de la influencia histórica mediante una red recurrente profunda (LSTM) permiten la obtención de mejores resultados que los de la literatura en la predicción de palabras mediante el procesamiento de texto. En esta línea, con una estrategia básica de inversión en el índice S&P 500, se obtiene una rentabilidad promedio mayor que trabajos previos en la misma área (Tabla 11 y Tabla 12). Un compilado de los resultados previos en esta área y de esta naturaleza han sido expuesto en el trabajo de Ding Xiao [DING, 2015] y Luss Ronny [LUSS, 2012]. Ver tabla 12.

Se puede afirmar que los resultados del presente trabajo de fin de carrera presentan una alternativa de solución al problema planteado.

Tabla 12: Comparación versus la literatura actual

	MCC	Precisión	Rentabilidad (USD)
Luss y Aspremont [LUSS, 2012]	0.0711	56.42%	8,694
Ding Xiao [DING, 2015]	0.4357	65.08%	16'785
Trabajo actual. (EB-LSTM)	0.39	69.05%	17'535

15.2. Trabajos futuros y recomendaciones

Durante la realización del proyecto, se han descubierto y considerado los siguientes puntos para trabajos posteriores que podrían optimizar los resultados obtenidos:

- ✓ Consideración de noticias por acción individual para predecir solamente el valor de acciones individuales
- ✓ Inclusión de otros factores de movimiento de los precios bursátiles para poder optimizar los resultados obtenidos solamente por el uso de noticias: análisis de

sentimiento, indicadores macroeconómicos, tendencias previas y noticias en diferentes idiomas

- ✓ Consideración de una mejor estrategia de inversión que se ajuste mejor al modelo en vez de una estrategia simple, y entrenamiento de la red neuronal para que se ajuste también a tal estrategia
- ✓ Evaluación del presente trabajo en los índices de otros mercados bursátiles.



16. Bibliografía

[BENGIO, 2013] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798-1828.

[BENGIO, 2015] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553).

[BISHOP, 2006] Bishop, C. M. (2006). *Pattern recognition. Machine Learning*, 128.

[BORDER, 2014] Border, John y Houpt, Evan (2014). *Stock Market For Beginners Book: Stock Market Basics Explained for Beginners Investing in the Stock Market*

[BOX, 1976] Box, G. E., & Jenkins, G. M. (1976). *Time series analysis: forecasting and control*, revised ed. Holden-Day.

[BURTON, 2007] Malkiel, Burton. G. (2007). *A random walk down Wall Street: the time-tested strategy for successful investing*. WW Norton & Company.

[CAO, 2001] Cao, L., & Tay, F. E. (2001). Financial forecasting using support vector machines. *Neural Computing & Applications*, 10(2), 184-192.

[CHEN, 2009] Schumaker, R. P., & Chen, H. (2009). Textual analysis of stock market prediction using breaking financial news: The AZFin text system. *ACM Transactions on Information Systems (TOIS)* .

[CNMV, 2015] Comisión Nacional del Mercado de Valores
2016. <http://www.cnmv.es/legislacion/legislacion/leyes/refunley2.htm>

[CORTES, 1995] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3).

[COLAH, 2015] Olah, C. (2015). Understanding LSTM Networks--colah's blog. *Colah. github. io*.

[COWIE, 1996] Cowie, J., & Lehnert, W. (1996). Information extraction. *Communications of the ACM*, 39(1).

[DAMODARAN, 1996] Damodaran, A. (1996). *Corporate finance*. Wiley.

[DEEPLARNING4J, 2016] Deep Learning for Java (2016). <https://deeplearning4j.org>.

[DENG, 2014] Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4).

[DIGRANDE, 2014] Digrande Sebastian, Nolan Tim y Borno Ruba (2014). Break Up or Shake Up: Get Ready for the Technology Industry Revolution
<https://www.transactionadvisors.com/insights/break-or-shake-get-ready-technology-industry-revolution>

[DING, 2015] Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015, July). Deep Learning for Event-Driven Stock Prediction. In *IJCAI* (pp. 2327-2333).

[DOWNES, 2010] Downes, J., & Goodman, J. E. (2010). *Dictionary of finance and investment terms*. Barron's Educational Series.

[FAMA, 1965] Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business*, 38(1), 34-105.

[GABOR, 2015] Angeli, Gabor., Premkumar, M. J., & Manning, C. D. (2015, July). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

[GALLEGO, 2015] Gallego, Antonio (2015). BUILDING A REAL-TIME STOCK PREDICTION ENGINE POWERED BY SPRING XD, APACHE GEODE AND SPARK ML

[GOOGLE, 2015] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484-489.

[GOOGLEFINANCE, 2017] Google Finance (2017). Google Finance for S&P 500.
<https://www.google.com.pe/search?q=s%26p+index&oq=s%26p+index&aqs=chrome..69i57j69i60j69i59j69i61j0l2.1450j0j7&sourceid=chrome&ie=UTF-8>

[GRAVES, 2008] Graves, Alex (2008). Supervised Sequence Labelling with Recurrent Neural Networks

[GRIN, 2009] Schlichting, T. (2013). *Fundamental Analysis, Behavioral Finance and Technical Analysis on the Stock Market*. GRIN Verlag.

[GUO, 2013] Guo, J. (2013). Backpropagation through time. *Unpubl. ms., Harbin Institute of Technology*.

[HAMILTON, 2006] W. P. Hamilton (2006). The Stock Market Barometer: A Study of Its Forecast Value Based on Charles H. Dow's Theory

[HARDER, 2010] Harder, S. (2010). The Efficient Market Hypothesis and its Application to Stock Markets.

[HINTON, 2012] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6).

[HOBSON, 2012] Hobson, R. (2012). *Shares Made Simple: A Beginner's Guide to the Stock Market*. Harriman House Limited.

[INTELLIJ, 2016] IntelliJ IDE (2016). <https://www.jetbrains.com/idea/>

[IPYTHON, 2016] Taotao, L. (2016). IPython-Dashboard. <https://pypi.python.org/pypi/IPython-Dashboard>

[JAVA, 2017] Java (2017). <https://www.oracle.com/java/index.html>

[JURAFSKY, 2014] Jurafsky, D., & Martin, J. H. (2014). *Speech and language processing* (Vol. 3). Pearson.

[KARPATHY, 2015] Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*.

[KIM, 2003] Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1).

[KIMOTO, 1990] Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990, June). Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 1-6). IEEE.

[KOGAN, 2009] Kogan, S., Levin, D., Routledge, B. R., Sagi, J. S., & Smith, N. A. (2009, May). Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 272-280). Association for Computational Linguistics.

[LASKER, 2011] Lasker, E. (2011). *Go and Go-moku*. Courier Corporation.

[LAI, 2014] Lai, P. F. B. W. C. H., & Hang, W. C. (2014). Performance of Stock Market Prediction. *Public Finance Quarterly*, 4, 471.

[LAWRENCE, 1997] Lawrence, R. (1997). Using neural networks to forecast stock market prices. *University of Manitoba*, 333.

[LAVRENKO, 2000] Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000, August). Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining* (pp. 37-44).

[LEBARON, 1999] LeBaron, B., Arthur, W. B., & Palmer, R. (1999). Time series properties of an artificial stock market. *Journal of Economic Dynamics and control*, 23(9), 1487-1516.

[LECUN, 1998] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

[LUSS, 2012] Luss, R., & d'Aspremont, A. (2012). Predicting abnormal returns from news using text classification. *Quantitative Finance*, 15(6).

[MATTHEWS, 1975] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2), 442-451.

[MCCORMINCK, 2016] McCorminck Chris (2016). Word2Vec Tutorial - The Skip-Gram Model. <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

[MARKETWATCH - 2015] Wallace, Witkowski (2015). Global stock market cap has doubled since QE's start.

[MEHRYAR, 2012] Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

[MIKOLOV, 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

[MNIH, 2013] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[MODI, 2016] Modi, A. (2016, August). Event embeddings for semantic script modeling. In *Proceedings of CoNLL*.

[NEELY, 1997] Neely, C., Weller, P., & Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? A genetic programming approach. *Journal of financial and Quantitative Analysis*, 32(04), 405-426.

[NETBEANS, 2016] Oracle (2016). NetBeans IDE - The Smarter and Faster Way to Code

[NVIDIA, 2016] Nvidia (2016). MACHINE LEARNING

[OFFICE, 2017] Microsoft (2017). Office Suite 2016 – Visio.
<https://products.office.com/en/visio/flowchart-software>

[OPENCSSV, 2017] Open CSV (2017). <http://opencsv.sourceforge.net/>

[PESARAN, 1994] Pesaran, M. H., & Timmermann, A. (1994). Forecasting stock returns An examination of stock market trading in the presence of transaction costs. *Journal of Forecasting*, 13(4).

[PREDICTWALLSTREET, 2016] Predict Wall Street Home Page (2016).
<https://predictwallstreet.com/>

[POWERS, 2011] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

[PYTHON, 2016] About Python (2016). About Python. <https://www.python.org/about/>

[SALTON, 1998] Salton, G. (1989). Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*.

[SCRAPY, 2017] Scrapy (2017). Intro to Scrapy. <https://scrapy.org/>

[SOCHER, 2013] Socher, R., Chen, D., Manning, C. D., & Ng, A. (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems* (pp. 926-934).

[STANDFOR, 2016] Stanford University (2016). Stanford Open Information Extraction. <http://nlp.stanford.edu/software/openie.html>

[TENSORFLOW, 2016] TensorFlow (2016). TensorFlow - an Open Source Software Library for Machine Intelligence. <https://www.tensorflow.org/>

[THAWORNWONG, 2004] Enke, D., & Thawornwong, S. (2004). The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with applications*, 29(4), 927-940.

[WAVENET, 2016] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... & Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. CoRR abs/1609.03499.

[WHITE, 1993] White, H. (1993). Economic prediction using neural networks: The case of IBM daily stock returns.

[WONG - 2007] Tsang, P. M., Kwok, P., Choy, S. O., Kwan, R., Ng, S. C., Mak, J., ... & Wong, T. L. (2007). Design and implementation of NN5 for Hong Kong stock price forecasting. *Engineering Applications of Artificial Intelligence*, 20(4), 453-461.

[WORD2VEC, 2017] Ling, W., Dyer, C., Black, A. W., & Trancoso, I. (2015). Two/Too Simple Adaptations of Word2Vec for Syntax Problems. In HLT-NAACL (pp. 1299-1304).

[YAHOO API, 2016] Yahoo Quotes (2016) .Quotes API for Yahoo Finance. <http://financequotes-api.com/>