



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE CIENCIAS

CARRERA DE INGENIERÍA EN ESTADÍSTICA INFORMÁTICA

“REDES NEURONALES ARTIFICIALES Y MODELOS ARIMA APLICADAS A LA MODELACIÓN Y PREDICCIÓN DEL TIPO DE CAMBIO EURO-DÓLAR”

Trabajo de titulación

TIPO: Proyecto de Investigación

Presentado para optar al grado académico de:

INGENIERO EN ESTADÍSTICA INFORMÁTICA

AUTOR: WILLIAN DANILO ALCOSER MANYA

DIRECTORA: Ing. JENNY PATRICIA PAREDES FIERRO

Riobamba – Ecuador

2020

© 2020, Willian Danilo Alcoser Manyá

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo cita bibliográfica del documento, siempre y cuando se reconozca el Derecho del Autor.

Yo, Willian Danilo Alcoser Manya, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 26 de agosto de 2020



Willian Danilo Alcoser Manya

060465682-7

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE CIENCIAS

CARRERA DE INGENIERÍA EN ESTADÍSTICA INFORMÁTICA

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación; tipo investigativo “REDES NEURONALES ARTIFICIALES Y MODELOS ARIMA APLICADAS A LA MODELACIÓN Y PREDICCIÓN DEL TIPO DE CAMBIO EURO-DÓLAR”, realizado por el señor: WILLIAN DANILO ALCOSER MANYA, ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. Nancy Elizabeth Chariguaman Maurisaca PRESIDENTE DEL TRIBUNAL	_____	2020-08-26
Ing. Jenny Patricia Paredes Fierro DIRECTORA DE TRABAJO DE TITULACIÓN	_____	2020-08-26
Dr. Jorge Washington Congacha Ausay MIEMBRO DEL TRIBUNAL	_____	2020-08-26

AGRADECIMIENTO

Agradezco a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO ya que me abrió las puertas para que pueda cumplir con uno de mis objetivos, a mi directora del Trabajo de Titulación Ing. Patricia Paredes por sus conocimientos y guía durante este proceso, miembro del tribunal Dr. Jorge Congacha, a los docentes quienes me supieron enseñar con paciencia y dedicación, y de manera especial a la Ing. Isabel Escudero ya que con su carisma y entusiasmo ha logrado motivar a más de uno a tener un amor por la Estadística.

Agradezco especialmente a mi padre José y madre María por su apoyo incondicional, a mis hermanos, mis sobrinos y compañeros de clase ya que de cada uno pude aprender algo para seguir avanzando.

Willian

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	viii
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE GRÁFICOS.....	x
ÍNDICE DE ANEXOS.....	xi
ÍNDICE DE ABREVIATURAS.....	xii
RESUMEN.....	xiii
ABSTRACT.....	xiv
INTRODUCCIÓN.....	1

CAPÍTULO I

1. MARCO REFERENCIAL.....	3
1.1. Antecedentes.....	3
1.2. Planteamiento del problema.....	5
1.2.1. <i>Enunciado del problema</i>	5
1.2.2. <i>Formulación</i>	5
1.3. Justificación.....	5
1.4. Objetivos.....	7
1.4.1. <i>Objetivo General</i>	7
1.4.2. <i>Objetivos Específicos</i>	7

CAPÍTULO II

2. MARCO TEÓRICO REFERENCIAL.....	8
2.1. Conceptos básicos.....	8
2.2. Procesos Estocásticos.....	10
2.2.1. <i>Proceso estocástico y serie temporal</i>	10
2.2.2. <i>Procesos estocásticos estacionarios</i>	10
2.2.2.1. <i>Función de autocovarianzas y de autocorrelación</i>	11
2.2.3. <i>Proceso Ruido Blanco</i>	13
2.3. Metodología Box-Jenkins o modelos ARIMA.....	14
2.3.1. <i>Modelamiento de series estacionarias</i>	14
2.3.1.1. <i>Modelos Autorregresivos – AR(p)</i>	14

2.3.1.2.	<i>Modelos de Media Móvil – MA(q)</i>	14
2.3.1.3.	<i>Modelos Mixtos Autorregresivos – Media Móvil, ARMA(p, q)</i>	15
2.3.2.	Modelamiento de series no estacionarias	15
2.3.2.1.	<i>Modelos ARIMA(p,d,q)</i>	15
2.3.2.2.	<i>Caminata aleatoria</i>	16
2.3.3.	<i>Fases</i>	16
2.3.4.	<i>Supuestos</i>	17
2.3.4.1.	<i>Independencia</i>	17
2.3.4.2.	<i>Normalidad</i>	17
2.3.4.3.	<i>Estacionariedad</i>	18
2.3.4.4.	<i>Homocedasticidad</i>	19
2.4.	Machine Learning	19
2.4.1.	<i>Aprendizaje Supervisado</i>	20
2.4.2.	<i>Aprendizaje No Supervisado</i>	21
2.4.3.	<i>Aprendizaje Semisupervisado</i>	21
2.4.4.	<i>Aprendizaje Por Refuerzo</i>	22
2.5.	Redes Neuronales	22
2.5.1.	<i>Elementos básicos</i>	24
2.5.1.1.	<i>Parámetros</i>	24
2.5.1.2.	<i>Tipos de funciones</i>	25
2.5.1.3.	<i>Aprendizaje de una red neuronal</i>	27
2.5.2.	<i>Tipos de redes neuronales</i>	29
2.6.	Redes neuronales recurrentes	29
2.6.1.	<i>Red de Elman</i>	31
2.6.2.	<i>Red de Jordan</i>	32
2.6.3.	<i>Red LSTM</i>	33
2.6.4.	<i>Red GRU</i>	36
2.7.	Medidas de evaluación de la precisión del pronóstico	37
2.7.1.	<i>Errores de pronóstico</i>	37
2.7.1.1.	<i>Errores dependientes de la escala</i>	37
2.7.1.1.1.	<i>Mean Square Error (MSE)</i>	37
2.7.1.1.2.	<i>Root Mean Square Error (RMSE)</i>	38
2.7.1.1.3.	<i>Mean Absolute Error (MAE)</i>	38
2.7.1.1.4.	<i>Median Absolute Error (MdAE)</i>	38
2.7.1.2.	<i>Porcentajes de error</i>	38
2.7.1.2.1.	<i>Mean Absolute Percentage Error (MAPE)</i>	38
2.7.1.2.2.	<i>Median Absolute Percentage Error (MdAPE)</i>	38

2.7.1.2.3.	<i>Root Mean Square Percentage Error (RMSPE)</i>	39
2.7.1.2.4.	<i>Root Median Square Percentage Error (RMdSPE)</i>	39
2.7.1.2.5.	<i>Symmetric Mean Absolute Percentage Error (sMAPE)</i>	39
2.7.1.2.6.	<i>Symmetric Median Absolute Percentage Error (sMdAPE)</i>	39
2.7.2.	<i>Coeficiente U de Theil</i>	40
2.7.3.	<i>Prueba de Diebold-Mariano</i>	40

CAPÍTULO III

3.	METODOLOGÍA	42
3.1.	Tipo de Investigación	42
3.2.	Diseño de la Investigación	43
3.2.1.	<i>Operacionalización de la variable</i>	43
3.2.2.	<i>Localización del Estudio</i>	43
3.2.3.	<i>Población de estudio</i>	43
3.2.4.	<i>Técnicas de recolección de datos</i>	44
3.2.5.	<i>Recolección de información</i>	44
3.2.6.	<i>Análisis Estadístico</i>	44

CAPÍTULO IV

4.	RESULTADOS Y DISCUSIÓN	46
4.1.	Aplicación ARIMA	48
4.2.	Aplicación Redes Neuronales Recurrentes	52
4.2.1.	<i>Célula Elman</i>	54
4.2.2.	<i>Célula LSTM</i>	55
4.3.	Comparación de las técnicas de predicción	57
4.3.1.	<i>Caracterización de las técnicas</i>	57
4.3.2.	<i>Medidas de evaluación de predicciones</i>	58
4.3.3.	<i>Análisis de precisión de los modelos</i>	58

	CONCLUSIONES	62
--	---------------------------	----

	RECOMENDACIONES	
--	------------------------------	--

644

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-3. Operacionalización de la variable en estudio.....	43
Tabla 1-4. Estadística descriptiva de la serie temporal EUR/USD	47
Tabla 2-4. Medidas de evaluación de predicciones	58
Tabla 3-4. Coeficiente U de Theil – predicciones con: ARIMA, Elman y LSTM	58
Tabla 4-4. Prueba de Diebold-Mariano “two.sided” y “greater” de las predicciones con: ARIMA, Elman y LSTM.....	59

ÍNDICE DE FIGURAS

Figura 1-2.	Funciones de autocorrelación	12
Figura 2-2.	Neurona biológica y Neurona artificial	23
Figura 3-2.	Estructura básica de una red neuronal artificial	24
Figura 4-2.	Neurona con dos entradas y una salida	26
Figura 5-2.	Funciones de activación habituales	27
Figura 6-2.	Neurona recurrente (izquierda), desarrollo en el tiempo (derecha)	30
Figura 7-2.	Capa de neuronas recurrentes (izquierda), desarrollo en el tiempo (derecha).	30
Figura 8-2.	Red Neuronal de Elman	32
Figura 9-2.	Red Neuronal de Jordan.....	33
Figura 10-2.	Red Neuronal LSTM	34
Figura 11-2.	Red Neuronal GRU.....	36

ÍNDICE DE GRÁFICOS

Gráfico 1-4.	Serie de la cotización diaria del tipo de cambio EUR/USD.....	46
Gráfico 2-4.	Histograma de la serie temporal EUR/USD	47
Gráfico 3-4.	Función de autocorrelación simple.....	48
Gráfico 4-4.	Prueba de Dickey-Fuller.....	48
Gráfico 5-4.	Prueba de Dickey-Fuller de la serie diferenciada	49
Gráfico 6-4.	Serie Integrada de orden 1 y ACF	49
Gráfico 7-4.	Diagnóstico del modelo.....	50
Gráfico 8-4.	Prueba de Dickey-Fuller.....	51
Gráfico 9-4.	Prueba de Box-Ljung	51
Gráfico 10-4.	Prueba de Lilliefors	51
Gráfico 11-4.	Prueba de Goldfeld-Quandt.....	51
Gráfico 12-4.	Valores Reales vs. Predichos – Modelo ARIMA	52
Gráfico 13-4.	Red con células de Elman	54
Gráfico 14-4.	Comportamiento del error de la red de Elman.....	54
Gráfico 15-4.	Valores reales vs. Predichos – RNR de Elman.....	55
Gráfico 16-4.	Red con células LSTM.....	55
Gráfico 17-4.	Comportamiento del error de la red LSTM	56
Gráfico 18-4.	Valores reales vs. Predichos – RNR LSTM	56
Gráfico 19-4.	Predicciones realizadas con: ARIMA, Elman y LSTM	57
Gráfico 20-4.	MAPE de las 22 predicciones	60

ÍNDICE DE ANEXOS

- ANEXO A:** Código en R (versión 3.5.3) del análisis exploratorio de la serie
- ANEXO B:** Código en R (versión 3.5.3) metodología Box-Jenkins
- ANEXO C:** Código en R (versión 3.5.3) Red neuronal con células de tipo ELMAN
- ANEXO D:** Código en Python (versión 2.7.13) de la red neuronal con células LSTM
- ANEXO E:** Código en R (versión 3.5.3) de comparación de las predicciones de las redes con células Elman y LSTM
- ANEXO F:** Código en R (versión 3.5.3) del Coeficiente U de Theil
- ANEXO G:** Código en R (versión 3.5.3) de la Prueba de Diebold-Mariano

ÍNDICE DE ABREVIATURAS

ANN: Artificial Neural Network

RNN: Recurrent Neural Networks

LSTM: Long Short-Term Memory

ARIMA: Autoregressive Integrated Moving Average

FOREX: Foreign Exchange

MSE: Mean Square Error

RMSE: Root Mean Square Error

MAE: Mean Absolute Error

MdAE: Median Absolute Error

MAPE: Mean Absolute Percentage Error

MdAPE: Median Absolute Percentage Error

RMSPE: Root Mean Square Percentage Error

RMdSPE: Root Median Square Percentage Error

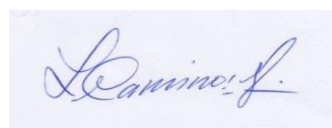
sMAPE: Symmetric Mean Absolute Percentage Error

sMdAPE: Symmetric Median Absolute Percentage Error

RESUMEN

El presente trabajo de titulación tiene como objetivo modelar y predecir el tipo de cambio Euro-Dólar, para lo cual se obtuvo los datos en la plataforma virtual Investing.com cuyo acceso es libre; se realizó un análisis exploratorio para identificar el comportamiento de los datos. Se aplicó tres técnicas las cuales son: Box-Jenkins, redes neuronales recurrentes de tipo Elman y *Long Short-Term Memory* para modelar y predecir la serie, se obtuvo los errores de escala y porcentuales, el coeficiente U de Theil y la prueba de Diebold-Mariano para identificar que técnica presenta predicciones más precisas. Los datos se obtuvieron en el periodo desde el 2 de enero de 1998 hasta el 31 de diciembre del 2019, se dividió la serie en conjuntos de entrenamiento y validación. Para el conjunto de entrenamiento se usó los datos del 2 de enero de 1998 al 29 de noviembre de 2019 y desde el 2 de diciembre de 2019 al 31 de diciembre del mismo año para el conjunto de validación. Debido a que la serie temporal presenta características de caminata aleatoria se determina un modelo ARIMA(0, 1, 0), cuya predicción en el tiempo t es el dato del tiempo t-1, se realizó predicciones de 22 días al futuro y se comparó con las predicciones realizadas con los dos tipos de redes neuronales recurrentes. Los coeficientes U de Theil para las predicciones con el modelo ARIMA, red de Elman y red *Long Short-Term Memory* dieron 0.04743, 0.002625 y 0.001808 respectivamente, identificando que la tercera técnica presenta pronósticos más precisos para este tipo de datos, pudiéndose corroborar mediante la prueba de Diebold-Mariano.

Palabras clave: <ESTADÍSTICA>, <REDES NEURONALES RECURRENTEES>, <METODOLOGÍA BOX-JENKINS>, <MODELAMIENTO DE SERIES ESTACIONARIAS>, <TIPO DE CAMBIO EURO-DÓLAR>



28-07-2020

0154-DBRAI-UPT

ABSTRACT

The present degree work aims to model and predict the EURO-DOLLAR exchange rate, for which the data was obtained on the virtual platform Investing.com whose access is free; an exploratory analysis was performed to identify the behavior of the data. Three techniques were applied: Box-Jenkins, Recurrent Elman-type recurrent neural networks *and Long Short-Term Memory* to model and predict the series, the scale and percentage errors, Theil's U coefficient and the Diebold-Mariano test were obtained to identify which technique has more accurate predictions. The data was obtained in the period from January 2, 1998 to December 31, 2019, the series was divided into training and validation sets. Data from January 2, 1998 to November 29, 2019 and from December 2, 2019 to December 31 of the same year for the validation set were used for the training package for The Training Set for January 29, 2019 and December 31, 2019. Because the time series has random walk characteristics an ARIMA(0, 1, 0) model is determined, the prediction in time t is the time data t-1, predictions of 22 days to the future were made and compared to predictions made with the two types of recurrent neural networks. The U coefficients of Theil for predictions with the model ARIMA, Elman network and Long Short-Term Memory network gave 0.04743, 0.002625 and 0.001808 respectively, identifying that the third technique presents more accurate forecasts for this type of data, being corroborated by the Diebold-Mariano test.

Keywords: <Statistics>, <Recurrent Neural Networks>, <Box-Jenkins Methodology>, <Stationary Series Modeling>, <Euro-Dollar Exchange Rate>.

INTRODUCCIÓN

Con la llegada del internet en el año de 1970 la economía mundial ha crecido a pasos agigantados debido a la globalización. Si bien es cierto los países intercambian sus bienes y servicios para mejorar su economía, también lo hacen para satisfacer alguna necesidad en particular. En la actualidad existen diversos mecanismos que facilitan las exportaciones de productos a personas de diferentes países del mundo, entre las cuales se puede identificar el transporte, la comunicación y los métodos de pago, originándose el mercado de divisas o también conocido como Forex.

Al igual que los precios de los demás bienes y servicios se establecen mediante la interacción de compradores y vendedores, los tipos de cambio vienen determinados por la interacción de los particulares, de las empresas y de las instituciones financieras, que compran y venden moneda extranjera con el fin de realizar pagos internacionales. La plataforma virtual en donde se puede comprar y vender monedas extranjeras se denomina Forex.

Forex es la abreviación de Foreign Exchange que significa intercambio de divisas. Es un mercado mundial descentralizado en donde se operan todas las divisas del mundo. Este mercado es el más grande en liquidez a nivel mundial, con un volumen diario de operaciones que supera los cinco mil millones de dólares. Los demás mercados bursátiles en el mundo en conjunto no se acercan a similares cantidades (FXCM). Forex funciona los 5 días de la semana, las 24 horas del día, se comercializa mediante plataformas virtuales con agentes o intermediarios financieros denominados brokers. Las entidades que participan en el intercambio de monedas son: bancos centrales, bancos comerciales, brokers y las multinacionales (Krygman and Obstfeld, 2006, p.332). Los activos que se negocian en Forex son índices de bolsa y mercancías como: oro, plata, cobre, petróleo, entre otros, y pares de divisas como: Euro-Dólar, Euro-Yen, Libra-Dólar, Libra-Yen, y otras.

El análisis del comportamiento futuro de los pares de divisas es de importancia para gobiernos, instituciones financieras e inversionistas, quienes utilizan para conocer la situación económica de un país, o para saber cuando vender y cuando comprar un bien o un servicio de un determinado lugar.

El presente trabajo está estructurado de la siguiente manera:

En el **capítulo I** se detalla los antecedentes del trabajo de investigación en donde se explica la evolución que ha tenido el mercado de divisas así como su origen, las diferentes técnicas que utilizan analistas financieros para estudiar el comportamiento futuro del mercado y se citan algunos ejemplos en donde se han aplicado redes neuronales artificiales y modelos ARIMA, se detalla también la problemática que se ha encontrado al momento de participar en el mercado de divisas, la justificación y por último los objetivos planteados en la investigación.

En el **capítulo II** se explica las técnicas estadísticas utilizadas para la modelación y predicción del tipo de cambio Euro-Dólar o EUR/USD. Los modelos utilizados para este estudio son: La metodología Box-Jenkins (conocidos también como modelos ARIMA), y Redes neuronales artificiales recurrentes (RNR), de tipo Elman y de tipo Long Short-Term Memory (LSTM). Los modelos ARIMA crean una ecuación matemática para predefinir el comportamiento de los datos temporales, y las RNR se basan en algoritmos que resuelven un problema en particular.

Los modelos ARIMA tienen una característica en particular; no necesitan de variables independientes para predecir o modelar las series de tiempo. En el caso de las redes neuronales artificiales (RNA) se trata de copiar el funcionamiento de las neuronas biológicas mediante algoritmos para crear modelos computacionales que ayuden a solucionar problemas. Su proceso de funcionamiento es entrenar a las neuronas para que realicen una acción específica (Pilco y Acurio 2019).

En el **capítulo III** hace referencia al marco metodológico en donde se detalla el diseño del trabajo de titulación como la recolección de datos, tipo de datos, población de estudio, localización del estudio y un breve resumen de la aplicación de las técnicas estadísticas.

En el **capítulo IV** se presenta los resultados obtenidos aplicando cada técnica de predicción utilizando métodos gráficos y analíticos, posteriormente se realiza una comparación de las técnicas y se escoge el modelo que tenga un alto grado de confiabilidad y poco error, para lo cual se ha utilizado el coeficiente U de Theil que mide la precisión de los datos pronosticados y la prueba de Diebold-Mariano que identifica el mejor modelo de acuerdo con los datos pronosticados. Luego se genera predicciones futuras a cinco días utilizando el modelo que mejor se ajuste a este tipo de datos. Además, se presenta conclusiones del estudio, recomendaciones, bibliografía y anexos (detalle de los algoritmos utilizados para la modelación y predicción del tipo de cambio).

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. Antecedentes

Al final de la segunda guerra mundial representantes de aproximadamente 45 países se reunieron con el fin de instaurar un nuevo orden económico mundial, el objetivo primordial era crear una base estable para los mercados que estaban devastados al finalizar la guerra, llevándolos a hacer un acuerdo llamado Bretton Woods. En virtud del acuerdo se estableció la creación del Fondo Monetario Internacional y el Banco Mundial (Historia del Forex).

En 1971, el acuerdo se rompió, surgiendo el sistema de cambio de divisas actual. Posteriormente en el año de 1980 con la introducción de nuevas tecnologías empezaron a propiciar la globalización del mercado de cambio de divisas y hacia los años 90 con la aparición y extensión del internet en todo el mundo el mercado Forex se convirtió en un mercado global, continuo y accesible para todos los inversores.

Desde entonces se ha venido creando formas de analizar el mercado para identificar los próximos movimientos, como es el caso del análisis fundamental, análisis técnico y análisis mixto. Además, existen técnicas estadísticas que modelan el comportamiento de los precios y generan predicciones futuras.

A principios de 1990 se ha venido realizando estudios de datos económicos o financieros aplicando redes neuronales artificiales teniendo un gran éxito como método de estimación y pronóstico en funciones no lineales que se encuentran constantemente en las finanzas (Arango, Cabrera Llanos y Herrera 2013). En la actualidad existen estudios en donde se aplica redes neuronales artificiales de diferentes tipos en la predicción de pares de divisas, acciones de la bolsa de valores de New York, temperaturas, demandas de energía eléctrica entre otras.

A finales de 1990 se diseñó una clase de redes neuronales basadas principalmente en el uso de la teoría de estabilidad de Liapunov para el desarrollo de las leyes de aprendizaje, las redes se les conoce como redes neuronales diferenciales o dinámicas (Poznyak, Sanchez y Yu 2001) (Ortiz-arango y Cruz-aranda 2011), cuyas aplicaciones han sido exitosas, como por ejemplo, Arango, Cabrera Llanos, & Herrera (2013) en su estudio realizan un pronóstico de los índices accionarios DAX y S&P 500 utilizando redes neuronales diferenciales llegando a resultados satisfactorios. Los autores llegan a la conclusión de que la técnica de predicción es una herramienta muy buena para la descripción del comportamiento detallado de una variable financiera. Del mismo modo Ortiz-arango & Cruz-aranda (2011) presenta un

estudio sobre el modelado del comportamiento del tipo de cambio peso-dólar utilizando redes neuronales diferenciales

Villada, Muñoz, & García-Quintero (2016) realizan un estudio sobre el pronóstico del precio del oro usando la red neuronal perceptrón multicapa, para ello utilizan cuatro neuronas en la capa de entrada las cuales son: el índice del dólar estadounidense, el índice S&P 500, los precios del petróleo y el precio del oro, se obtuvo mejores resultados al eliminar el índice del dólar estadounidense en el análisis.

Zapata & Hugo (2008) plantea doce variables económicas que sirven como entrada a un sistema de redes neuronales, su trabajo se basa en identificar que variable tiene mayor significancia en la predicción del tipo de cambio peso-dólar.

En la última década las redes neuronales artificiales de tipo Long Short Term Memory han tenido una gran popularidad al predecir datos de tipos secuenciales, su mecanismo para guardar información de largo y corto plazo lo ha hecho diferente y potente al realizar predicciones de datos históricos. La red se ha aplicado para la comprensión de textos de lenguajes naturales, reconocimiento de escritura a mano, reconocimiento de voz, reconocimiento de gestos, captura de imágenes (Calvo 2018). A continuación, se presenta algunos estudios de aplicaciones de las redes LSTM como técnica de predicción que involucran sistemas no lineales:

Granel & Salvador (2017) combina las redes neuronales LSTM con la estrategia de trading Bandas de Bollinger para clasificar compras, ventas y mantener, el cual utiliza el precio de cierre de las acciones de las 500 empresas que conforman el índice S&P 500; Dancho (2018) predice manchas solares en un tiempo de diez años; Cabezón Manchado (2018) compara las redes de tipo Elman, Jordan, LSTM y GRU en la predicción de la demanda eléctrica española; Pilco y Acurio (2019) en su trabajo de tesis denominado Técnicas estadísticas para la modelación y predicción de la temperatura y velocidad de viento en la provincia de Chimborazo, comparan los modelos ARIMA, la Teoría del Caos y la red neuronal de tipo Elman; Reyes Lora (2016) realiza una comparación de los modelos ARIMA con redes neuronales para predecir la demanda eléctrica de España.

En el presente estudio se analiza el tipo de cambio EUR/USD y se predice a 22 días a futuro, se compara los resultados obtenidos en las predicciones utilizando las técnicas de predicción ARIMA y redes neuronales recurrentes de tipo Elman y LSTM. Se obtiene los errores de pronóstico, el coeficiente U de Theil y la prueba de Diebold-Marino para identificar el mejor modelo.

1.2. Planteamiento del problema

1.2.1. Enunciado del problema

Operar en el mercado de divisas conlleva mucha disciplina, esfuerzo y preparación para saber cuándo comprar o vender un activo. Existen tres técnicas de análisis que ayudan a los inversionistas tomar una mejor decisión, entre estas tenemos: análisis fundamental, análisis técnico y análisis mixto, este último es la combinación del primero y segundo. En el análisis fundamental se basa en noticias macroeconómicas de un país como crecimiento de la economía, inflación, flujo de capital, etc. Murphy J. (2000, p-27) define el análisis técnico como el estudio de los movimientos del mercado, principalmente mediante el uso de gráficos, con el propósito de pronosticar las futuras tendencias de los mercados. Existen herramientas y técnicas estadísticas que explican el comportamiento pasado de los índices de precio.

Existe un dato estadístico muy conocido entre los especuladores del mercado Forex, donde se afirma que el 95% de los operadores pierden su dinero, aunque no existen estudios que corroboren dicha información, según el portal financiero Trading United en un artículo comenta lo siguiente: *“Incluso cuando no tenemos nada certero y conclusivo que respalde que el 95% de los traders minoristas pierde su dinero, es bastante seguro decir que un alto porcentaje de traders en Forex pierde dinero”* (UNITED). Esto ocurre debido al descontrol emocional, a la falta preparación por parte del operador y a indicadores estadísticos no muy efectivos en la predicción del mercado.

1.2.2. Formulación

En los sitios web se puede identificar que existen academias que enseñan a comercializar en el mercado Forex, cada una tiene su propia estrategia que integran indicadores para saber lo que puede hacer el precio en el futuro pero si bien es cierto estos indicadores tienen un efecto rezagado y las estrategias tienen porcentajes bajos de aciertos. Esto a llevado a que un alto porcentaje de personas pierdan su dinero en el mercado de divisas, por tal razón la pregunta a resolver es la siguiente:

¿Las técnicas propuestas predicen el tipo de cambio EUR/USD con un alto grado de confiabilidad?

1.3. Justificación

A medida que la tecnología avanza se ha creado diferentes plataformas digitales en donde las personas de diferentes partes del mundo pueden generar dinero y a medida que pasa el tiempo más

personas tienen accesibilidad a estos recursos. Una de estas plataformas es Forex cuya función es intercambiar o comercializar divisas con personas en cualquier parte del mundo. Su rápido y fácil acceso al realizar las operaciones en tiempo real ha hecho que gran cantidad de personas vean al mercado como una ola llena de oportunidades a la hora de generar ingresos. Se puede participar en el mercado de divisas las veinte y cuatro horas del día, cinco días de la semana utilizando un dispositivo móvil o laptop con conexión a internet

Los bancos centrales a través del sistema financiero de sus países, facilitan sus transacciones, ya que cuenta con una bolsa variada de monedas, la diversificación con la que cuentan permite que si una de ellas se devalúa, la otra se aprecie por lo que se compensa y protegen de los vaivenes del mercado (Terán, Gonzáles and Mosquera, 2011, p-6).

Por su gran número de operaciones que se realizan diariamente los precios de las divisas tienen un comportamiento muy volátil, esto ha llevado a que el porcentaje de operadores que pierden dinero en Forex se incremente en comparación con el porcentaje de operadores que ganan dinero. Existen diferentes métodos o estrategias para operar en el mercado constituidas con indicadores (gran parte de ellos estadísticos) que “ayudan” a identificar el próximo movimiento del mercado, pero la mayoría de ellos no tienen un alto grado de confiabilidad por lo que los operadores no poseen rendimientos.

Por tal motivo el objetivo del presente trabajo es predecir el tipo de cambio EUR/USD para identificar el posible movimiento del mercado después de cinco días. Para ello se utilizará las técnicas: ARIMA, y redes neuronales artificiales recurrentes de tipo ELMAN y LSTM. Se tomaron en cuenta estas técnicas debido a que son capaces de predecir series temporales y por qué en la actualidad existen estudios en donde aplican redes neuronales recurrentes y diferenciales llegando a resultados satisfactorios y con alto grado de confiabilidad.

Este proyecto de investigación beneficiará a los inversionistas particulares, empresas gestoras de fondos de inversión, compañías aseguradoras, las instituciones financieras no bancarias, turistas y estudiantes que vayan a tener una estadía en uno de los países en donde se emite las monedas euro y dólar americano.

1.4. Objetivos

1.4.1. *Objetivo General*

Predecir el tipo de cambio Euro-Dólar utilizando los modelos ARIMA y Redes Neuronales Artificiales Recurrentes.

1.4.2. *Objetivos Específicos*

- Realizar un análisis exploratorio de la serie.
- Construir modelos de predicción mediante el modelo ARIMA y redes neuronales recurrentes de tipo ELMAN y LSTM.
- Comparar los modelos obteniendo las medidas de evaluación de pronósticos, coeficiente U de Theil y realizando la prueba de Diebold-Mariano.
- Identificar los días en las que la técnica seleccionada es eficaz.

CAPÍTULO II

2. MARCO TEÓRICO REFERENCIAL

2.1. Conceptos básicos

Forex

Forex es la conversión de una divisa a otra, es uno de los mercados más activos del mundo. Un gran número de operaciones se llevan a cabo por razones prácticas como el pago a trabajadores de empresas multinacionales, compra y venta de bienes o servicios, viajes a otros países por razones de turismo o educación entre otras, pero la gran conversión de divisas se realiza con la finalidad de obtener beneficios.

Es un mercado descentralizado es decir no existe un lugar físico o una sede, las operaciones se lo realiza mediante plataformas virtuales, es un mercado apalancado es decir que se puede operar en el mercado con poca cantidad de dinero, existen brokers que permiten operar con un mínimo de cinco dólares americanos, está abierto de lunes a viernes las veinticuatro horas del día.

A las personas o entidades que participan en cualquier mercado financiero mediante los servicios de un bróker se les conoce como comerciantes o traders, dicha actividad se conoce como “trading”.

Un bróker es una entidad financiera o persona encargada de colocar órdenes de compra y venta de acciones que cotizan en la bolsa o de valores financieros.

Principales divisas

Las principales divisas que se cotizan en el mercado Forex se listan a continuación:

- Euro – EUR
- Dólar Estadounidense – USD
- Libra Esterlina – GBP
- Yen Japonés – JPY
- Dólar Canadiense – CAD
- Dólar Australiano – AUD
- Franco Suizo – CHF
- Dólar Neozelandés – NZD

Los operadores que intervienen en el mercado de divisas hacen operaciones con pares de divisas, es decir, con los valores de una divisa con respecto a la otra.

Pares de divisas

Los principales pares del mercado de divisas (conocidos como Majors) son:

- EUR/USD
- GBP/USD
- USD/CAD
- USD/CHF
- AUD/USD
- USD/JPY

Un pip es el cambio mínimo que puede haber entre dos divisas (un par). Ejemplo: El EUR/USD cotiza a 1.1502 en un momento dado, y un minuto después cotiza a 1.1503, el precio se ha movido un pip. Otro ejemplo: GBP/USD cotiza a 1.3120 y un día después, el precio se ha movido a 1.3202, el precio se ha movido 82 pips ($1.3202 - 1.3120 = 0.0082$).

En pares como EUR/USD o GBP/USD se usa el cuarto decimal como la unidad mínima de cambio.

En base al número de pips que se haya movido un par de divisas que hayamos comprado o vendido, podremos calcular la ganancia o la pérdida de nuestra operación.

Participantes del mercado de divisas

Los individuos que intercambian divisas de un país por el otro son los participantes del mercado. Según Krugman & Obstfeld (2006, pp.331), los agentes o participantes en el mercado de divisas son los bancos comerciales, las empresas multinacionales que participan en el comercio internacional, las instituciones financieras no bancarias que gestionan activos, como las empresas gestoras de fondos de inversión y las compañías aseguradoras, los bancos centrales e inversores particulares.

- **Bancos Comerciales:** Los bancos comerciales se encuentran en el centro del mercado de divisas, porque la gran mayoría de las transacciones en el mercado de divisas se realiza a través del intercambio de depósitos bancarios denominados en diferentes monedas.
- **Las multinacionales:** Las empresas que operan en varios países suelen efectuar pagos o recibir ingresos en monedas diferentes de los países en los que están establecidos.
- **Las instituciones financieras no bancarias:** A lo largo de los años, la liberación de los mercados financieros en estados Unidos, Japón y otros países han fomentado la diversificación de los servicios ofrecidos por las instituciones financieras no bancarias a sus clientes, de los cuales la mayor parte coincide con las operaciones ofrecidas por los bancos.
- **Los bancos centrales:** Son agentes de carácter público que operan con mayor regularidad en el mercado de divisas.
- **Estudiantes y turistas que viajan al país donde se emite la moneda.**

(Krugman y Obstfeld 2006)

2.2. Procesos Estocásticos

Una serie temporal univariante consiste en un conjunto de observaciones de una variable de interés. Un modelo de series temporales debe reproducir las características de la serie. Si contamos con T observaciones, Y_t , $t = 1, 2, \dots, T$, el modelo univariante de series temporales se formulará en términos de los valores pasados de Y_t y/o su posición en relación con el tiempo (González 2007).

2.2.1. Proceso estocástico y serie temporal

Un proceso estocástico es una familia de variables aleatorias que, en general, están relacionadas entre sí y siguen una ley de distribución conjunta. Se denota por:

$$\dots, Y_{t-2}, Y_{t-1}, Y_t, Y_{t+1}, Y_{t+2}, \dots$$

En el marco estadístico de los procesos estocásticos, una serie temporal, Y_1, Y_2, \dots, Y_T , se puede interpretar como una realización muestral de un proceso estocástico que se observa únicamente para un número finito de periodos, $t = 1, 2, \dots, T$. En este contexto, una serie temporal es una sucesión de observaciones en la que cada una de ellas corresponde a una variable aleatoria distinta, y a la ordenación de la sucesión de observaciones es esencial para el análisis de la misma.

(González 2007)

2.2.2. Procesos estocásticos estacionarios

El objetivo de realizar análisis de series temporales es utilizar la teoría de procesos estocásticos para determinar qué proceso estocástico ha sido capaz de generar la serie temporal bajo estudio con el fin de caracterizar el comportamiento de la serie y predecir en el futuro. Dentro de la teoría de la predicción es siempre la misma: se aprende de las regularidades del comportamiento pasado de la serie y se proyectan hacia el futuro. Por lo tanto, es preciso que los procesos estocásticos generadores de las series temporales tengan algún tipo de estabilidad. Si, por el contrario, en cada momento de tiempo presentan un comportamiento diferente o inestable, no se pueden utilizar para predecir. A estas condiciones que se les impone a los procesos estocásticos para que sean estables para predecir, se les conoce como estacionariedad.

El concepto de estacionariedad se puede caracterizar bien en términos de la función de distribución o de los momentos del proceso. En el primer caso, se hablará de estacionariedad en sentido estricto y, en el segundo, de estacionariedad de segundo orden o covarianza.

Estacionariedad estricta. - Un proceso estocástico, Y_t , es estacionario en sentido estricto si y solo si:

$$F[Y_{t_1}, Y_{t_2}, \dots, Y_{t_n}] = F[Y_{t_1+k}, Y_{t_2+k}, \dots, Y_{t_n+k}] \quad \forall (t_1, t_2, \dots, t_n) \text{ y } k$$

Es decir, si la función de distribución de cualquier conjunto finito de n variables aleatorias del proceso no se altera si se desplaza k periodos en el tiempo.

Estacionariedad en covarianza. – Un proceso estocástico; Y_t , es estacionario en covarianza si y solo si:

- Es estacionario en media, es decir, todas las variables aleatorias del proceso tienen la misma media y es finita:

$$E(Y_t) = u < \infty, \quad \forall t$$

- Todas las variables aleatorias tienen la misma varianza y es finita, es decir, la dispersión en torno a la media constante a lo largo del tiempo es la misma para todas las variables del proceso.

$$V(Y_t) = E[Y_t - u]^2 = \sigma_Y^2 < \infty, \quad \forall t$$

- Las autocovarianzas solo dependen del número de periodos de separación entre las variables y no del tiempo, es decir, la covarianza lineal entre dos variables aleatorias del proceso que dicen k periodos de tiempo es la misma que existe entre cualesquiera otras dos variables que estén separadas también k periodos, independientemente del momento concreto de tiempo al que estén referidas

$$\text{cov}(Y_t, Y_s) = E[Y_t - u][Y_s - u] = \gamma_{|t-s|} = \gamma_k < \infty, \quad \forall t$$

Por lo tanto, un proceso estocástico es estacionario si y solo sí;

- $E(Y_t) = u < \infty$
- $\text{Cov}(Y_t, Y_s) = \begin{cases} V(Y_t) = \sigma_Y^2 < \infty & t = s \\ \gamma_{|t-s|} = \gamma_k < \infty & t \neq s \end{cases}$

(González 2007)

2.2.2.1. Función de autocovarianzas y de autocorrelación

Función de autocovarianzas. - La función de autocovarianzas de un proceso estocástico estacionario es una función de k (número de periodos de separación entre las variables) que recoge el conjunto de autocovarianzas del proceso y se denota por:

$$\gamma_k, \quad k = 0, 1, 2, 3, \dots$$

La función de autocovarianzas de un proceso estocástico recoge toda la información sobre la estructura dinámica lineal del mismo. Pero depende de las unidades de medida de la variable, por lo que, en general, se suele utilizar la función de autocorrelación.

Las características de la función de autocovarianzas son:

- Incluye la varianza del proceso para $k = 0$: $\gamma_0 = E[Y_t - u][Y_t - u] = V(Y_t)$
- Es una función simétrica: $\gamma_k = E[Y_t - u][Y_{t+k} - u] = E[Y_t - u][Y_{t-k} - u] = \gamma - k$

(González 2007)

Función de autocorrelación. - El coeficiente de autocorrelación de orden k de un proceso estocástico estacionario mide el grado de asociación lineal existente entre dos variables aleatorias del proceso separadas k periodos:

$$\rho_k = \frac{cov(Y_t, Y_{t+k})}{\sqrt{V(Y_t)V(Y_{t+k})}} = \frac{\gamma_k}{\sqrt{\gamma_0\gamma_0}} = \frac{\gamma_k}{\gamma_0}$$

Por ser un coeficiente de correlación, no depende de unidades y $|\rho_k| \leq 1, \forall k$.

La función de autocorrelación de un proceso estocástico estacionario es una función de k que recoge el conjunto de los coeficientes de autocorrelación del proceso y se denota por $\rho_k, k = 0, 1, 2, 3, \dots$

La función de autocorrelación se suele representar gráficamente por medio de un correlograma como se muestra a continuación:

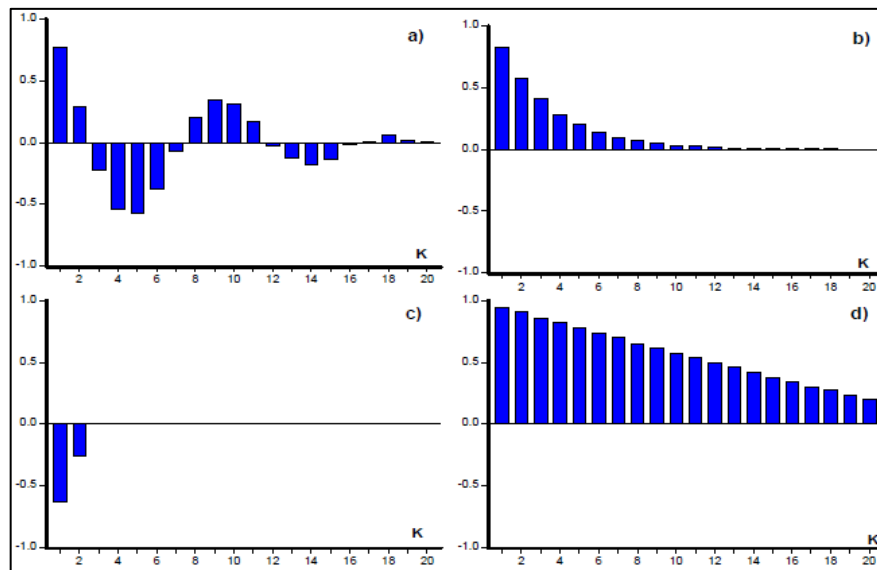


Figura 1-2. Funciones de autocorrelación

Fuente: (González 2007)

En la figura 1-2 muestran 4 correlogramas correspondientes a diferentes series temporales. Los correlogramas a), b) y c) decrecen rápidamente hacia cero conformes aumenta k : exponencialmente en los casos a) y b) y truncándose en el caso c). Son, por lo tanto, correlogramas correspondientes a series estacionarias. Por el contrario, los coeficientes de autocorrelación del correlograma d) decrecen lentamente, de forma lineal, por lo que no corresponden a una serie estacionaria.

Las características de la función de autocorrelación de un proceso estocástico estacionario son:

- El coeficiente de autocorrelación de orden 0 es, por definición 1. Por eso, a menudo, no se le incluye explícitamente en la función de autocorrelación.

$$\rho_0 = \frac{\gamma_0}{\gamma_0} = 1$$

- Es una función simétrica: $\rho_k = \frac{\gamma_k}{\gamma_0} = \frac{\gamma_{-k}}{\gamma_0} = \rho_{-k}$. Por ello, en el correlograma se representa la función de autocorrelación solamente para los valores positivos del retardo k .
- La función de autocorrelación de un proceso estocástico estacionario tiende a cero rápidamente cuando k tiende a ∞ .

(González 2007)

2.2.3. Proceso Ruido Blanco

El proceso estocástico más sencillo es el denominado Ruido Blanco que es una secuencia de variables aleatorias de media cero, varianza constante y covarianzas nulas.

Habitualmente se denota de la siguiente manera:

$$\varepsilon_t, \quad t = 0, \pm 1, \pm 2, \dots$$

$$E(\varepsilon_t) = 0, \quad \forall t \quad V(\varepsilon_t) = \sigma^2, \quad \forall t \quad Cov(\varepsilon_t \varepsilon_s) = 0, \quad \forall t \neq s$$

Un proceso de Ruido Blanco, $\varepsilon_t \sim RB(0, \sigma^2)$, es estacionario si la varianza σ^2 es finita con función de autocovarianzas (FACV):

$$\gamma_k = \sigma^2, \quad k = 0, \quad y \quad \gamma_k = 0, \quad k > 0$$

Y función de autocorrelación (FAC):

$$\rho_k = 1, \quad k = 0, \quad y \quad \rho_k = 0, \quad k > 0$$

(González 2007)

2.3. Metodología Box-Jenkins o modelos ARIMA

El modelo ARIMA, es una técnica de predicción presentada por primera vez en la publicación *Time Series Analysis: Forecasting and Control*, en el año 1976 por Box y Jenkins para el modelamiento y predicción de series univariantes, es decir, que no requiere de variables regresoras para realizar una predicción si no únicamente la información contenida en los valores pasados. Los componentes se obtienen realizando las funciones de autocorrelación simple y parcial, además de realizar diferencias de orden n ($n > 1$) en caso de que la serie no sea estacionaria.

El interés de esta técnica de pronóstico consiste en realizar un análisis de las propiedades probabilísticas o estocásticas, de las series de tiempo económicas (Gujarati 2009).

El modelo ARIMA presenta una ecuación implícita que permite describir un valor como una función lineal de datos anteriores y errores debido al azar, puede incluir un componente cíclico o estacional. El objetivo consiste en obtener un modelo adecuado pero parsimonioso. Para aplicar el modelo ARIMA, Box y Jenkins recomiendan como mínimo unas 50 observaciones y estudiar las características esenciales de la serie como tendencia, homocedasticidad, estacionalidad, estacionariedad y funciones de autocorrelación. (César 2016)

2.3.1. Modelamiento de series estacionarias

2.3.1.1. Modelos Autorregresivos – $AR(p)$

Un modelo autorregresivo (AR) describe una clase particular de modelo dinámico en el que las observaciones en un momento dado son predecibles a partir de las observaciones previas del proceso más un término del error. El caso más simple es el $ARIMA(1,0,0)$ o $AR(1)$ o de primer orden, cuya expresión matemática es:

$$AR(1) \equiv Y_t = \phi_1 Y_{t-1} + \varepsilon_t$$

El modelo autorregresivo de orden p , representado por $ARIMA(p, 0, 0)$, o $AR(p)$ toma la forma:

$$AR(p) \equiv Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t$$

Donde ϕ_1, \dots, ϕ_p son coeficientes a estimar y ε_t es un error aleatorio no correlacionado con media cero y varianza constante σ^2 .

(César 2016)

2.3.1.2. Modelos de Media Móvil – $MA(q)$

Un modelo de medias móviles (MA) es otro tipo de modelo dinámico donde el valor actual puede predecirse a partir de la componente aleatoria de este momento y, en menor medida, de los impulsos

aleatorios anteriores. El modelo $ARIMA(0, 0, 1)$, también denotado por $MA(1)$, viene representado por la expresión:

$$MA(1) \equiv Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

El modelo de medias móviles de orden q , representado por $ARIMA(0, 0, q)$, o también por $MA(q)$, viene dado por:

$$MA(q) \equiv Y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_{t-q}$$

Donde $\theta_1, \dots, \theta_q$ son coeficientes a estimar y ε es un error estocástico o ruido blanco.

(César 2016)

2.3.1.3. Modelos Mixtos Autorregresivos – Media Móvil, $ARMA(p, q)$

Una extensión natural de los modelos $AR(p)$ y $MA(q)$ es un tipo de modelos dinámicos que incluyen tanto términos autorregresivos como de medias móviles y se definen como $ARMA(p, q)$ o también como $ARIMA(p, 0, q)$. Su expresión es:

$$ARMA(p, q) \equiv Y_t = \phi_1 Y_{t-1} + \dots + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \dots - \theta_q \varepsilon_{t-q}$$

(César 2016)

2.3.2. Modelamiento de series no estacionarias

2.3.2.1. Modelos $ARIMA(p, d, q)$

Los modelos de las series no estacionarias se llaman modelos autorregresivos integrados de promedio móvil y se denota por $ARIMA(p, d, q)$. Donde \mathbf{p} indica el orden de la parte autorregresiva, \mathbf{d} indica el número de diferenciaciones, y \mathbf{q} indica el orden de la parte de promedio móvil.

- **Operador de retardo.** – El operador de retardo, B , retrasa una posición la observación. Así, por ejemplo: $BY_t = Y_{t-1}$, es decir, si B aplicada a Y_t retrasa ésta y la convierte en Y_{t-1} .
Aplicando el operador k veces obtenemos: $B^k Y_t = Y_{t-k}$.
- **Operador diferencias.** – Cuando una serie tiene tendencia es preciso quitarla mediante una diferencia regular. Una diferencia regular le resta a cada elemento de la serie la observación anterior: $w_t = Y_t - Y_{t-1}$.

La expresión puede escribirse:

$$w_t = Y_t - Y_{t-1} = Y_t - BY_t = (1 - B)Y_t$$

es decir, que tomar una diferencia equivale a multiplicar Y_t por $(1 - B)$. Si se toman d diferencias de orden 1, se representa por $(1 - B)^d Y_t$.

El modelo general denominado proceso autorregresivo integrado de promedio móvil de orden p , d , q , toma la siguiente expresión:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d Y_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) \varepsilon_t$$

El modelo permite describir una serie de observaciones después de que hayan sido diferenciadas d veces, a fin de extraer las posibles fuentes de no estacionariedad. Si hay algún patrón p , d , q igual a cero, se elimina el término correspondiente de la fórmula general.

(César 2016)

2.3.2.2. Caminata aleatoria

Un proceso estocástico univariante no estacionario Y_t es un proceso aleatorio cuando

$$Y_t = \mu + Y_{t-1} + \varepsilon_t \quad \forall t = 0, \pm 1, \pm 2, \dots$$

donde μ es un parámetro (que en muchas ocasiones vale 0) y $\varepsilon_t \sim RB(0, \sigma^2)$.

En un modelo de caminata aleatoria, la varianza de Y_t aumenta a través del tiempo, lo que es propia de un proceso no estacionario (Ríos 2008) (González 2007).

En el modelo de caminata aleatoria el valor de Y en el momento t es igual a su valor en el momento $t - 1$ mas una perturbación aleatoria. El modelo de caminata aleatoria es simplemente un modelo $AR(1)$ con parámetro $\phi = 1$:

$$Y_t = \phi_1 Y_{t-1} + \varepsilon_t$$

$$Y_t - \phi_1 Y_{t-1} = \varepsilon_t$$

$$Y_t - \phi_1 B Y_t = \varepsilon_t$$

$$(1 - \phi_1 B) Y_t = \varepsilon_t$$

$$(1 - B) Y_t = \varepsilon_t$$

2.3.3. Fases

Según Fernández (2016) las fases de la metodología Box y Jenkins son cuatro:

- **Fase 1:** En esta fase se identifica el posible modelo ARIMA que sigue la serie, lo que requiere: Convertir la serie observada en una serie estacionaria utilizando d diferencias y determinar los parámetros p y q

- **Fase 2:** Consiste en seleccionar provisionalmente un modelo para la serie estacionaria, se pasa a la segunda etapa de estimación, donde los parámetros AR y MA se estiman por máxima verosimilitud y se obtienen sus errores estándar y los residuos del modelo.
- **Fase 3:** En esta etapa se realiza la validación del modelo o el diagnóstico. Aquí se realiza la verificación de los supuestos como son: Independencia, Normalidad, Estacionariedad y Homocedasticidad.
- **Fase 4:** Aquí se realiza las predicciones con el modelo adecuado.

2.3.4. Supuestos

2.3.4.1. Independencia

Se realiza una prueba de hipótesis para probar si la variable aleatoria “residuos” son independientes o no, para ello se utiliza la prueba de Box-Ljung. Esta prueba se utiliza comúnmente en un modelo autorregresivo integrado de promedio móvil (Pilco y Acurio 2019). La hipótesis a probar es:

$$H_0: \rho_1 = \rho_2 = \dots = \rho_m = 0$$

$$H_1: \rho_i \neq 0 ; \forall i = 1, \dots, m$$

El estadístico de prueba se define de la siguiente manera:

$$Q = n(n+2) \sum_{k=1}^m \frac{r(k)^2}{n-k} \sim X_{m-1}^2$$

$$p = P(X_{m-1}^2 \geq Q)$$

Se plantea un nivel de significancia (por lo general se utiliza 0.05) y se rechaza la hipótesis nula cuando p es menor que el nivel de significancia (α).

En el software R se utiliza la función `Box.test()` con el parámetro `type = "Ljung-Box"`, para obtener el estadístico de prueba y el valor p de Ljung-Box.

2.3.4.2. Normalidad

Este supuesto de normalidad se verifica aplicando la prueba de Lilliefors que es una prueba de corrección de Kolmogorov-Smirnov (K-S). La prueba K-S para una muestra no es muy útil en la práctica, ya que la mayoría de las veces desconocemos cual es la media y la desviación estándar poblacional, y, por tanto, se deben estimar para la distribución teórica de comparación. Esto genera que la prueba K-S sea muy conservadora, aceptando la hipótesis nula en la mayoría de las ocasiones. La prueba de Lilliefors soluciona este problema al tabular el estadístico K-S para el caso en el que desconocemos la media y la desviación estándar poblacional y se estima a través de los datos muestrales (Romero Saldaña 2016).

La prueba de hipótesis a contrastar es:

H_0 : La variable aleatoria x se distribuye normalmente

H_1 : La variable aleatoria x no se distribuye normalmente

Con la función `lillie.test()` se obtiene el estadístico de prueba y el valor p en el software R, donde, si el valor p es menor que el nivel de significancia se rechaza H_0 .

2.3.4.3. Estacionariedad

Para verificar el supuesto de estacionariedad se aplica la prueba de Dickey-Fuller (DF), la prueba se basa en asumir que la serie se puede aproximar por un proceso $AR(1)$ con tres variantes: media cero, media diferente de cero y tendencia lineal. Inicialmente se asume que y_t sigue un modelo $AR(1)$ y se procede a transformar el modelo de la siguiente manera:

$$Y_t = \varphi_1 Y_{t-1} + \varepsilon_t$$

$$Y_t - Y_{t-1} = \varphi_1 Y_{t-1} - Y_{t-1} + \varepsilon_t$$

$$Y_t - Y_{t-1} = (\varphi_1 - 1)Y_{t-1} + \varepsilon_t$$

$$\Delta Y_t = \rho Y_{t-1} + \varepsilon_t$$

Donde $\rho = \varphi_1 - 1$. La existencia de una raíz unitaria equivale a $\varphi_1 = 1$, es decir, a $\rho = 0$.

1. Prueba DF para el caso 1: suponiendo que $Y_t \sim AR(1)$ con media cero, entonces

$$\Delta Y_t = \rho Y_{t-1} + \varepsilon_t$$

La hipótesis nula es $H_0: \rho = 0$ versus la alterna $H_1: \rho < 0$. El estadístico de la prueba se denota por τ y su distribución bajo H_0 permite calcular los valores críticos, de tal forma que el criterio de rechazo es $\tau < \tau_\alpha$, siendo τ_α el valor crítico relevante para la prueba DF.

$$\tau = \frac{\hat{\rho}}{\sigma_{\hat{\rho}}}$$

donde $\hat{\rho}$ es el coeficiente estimado y $\sigma_{\hat{\rho}}$ es el error estándar en el coeficiente estimado.

2. Prueba DF para el caso 2: suponiendo que $Y_t \sim AR(1)$ con media diferente de cero, entonces:

$$\Delta Y_t = \alpha + \rho Y_{t-1} + \varepsilon_t$$

donde α es una constante.

La hipótesis es la misma que la planteada en el caso 1.

3. Prueba DF para el caso 3: suponiendo que $Y_t \sim AR(1)$ con tendencia lineal, entonces:

$$\Delta Y_t = \alpha + \beta t + \rho Y_{t-1} + \varepsilon_t$$

donde α es una constante y β es un coeficiente en una tendencia de tiempo.

La hipótesis es la misma que la planteada en el caso 1.

(Giraldo Gomez 2012)

En el software R se utiliza la función *adf.test()* con la alternativa "stationary", para realizar la prueba.

2.3.4.4. Homocedasticidad

Se aplica la prueba de Quandt Goldfeld para verificar el supuesto de homocedasticidad. Para realizar esta prueba se divide la serie en dos grupos, y la evidencia de heterocedásticidad se basa en una comparación de la suma residual de cuadrados (RSS) usando el *F-estadístico*. Aunque Goldfeld y Quandt describieron dos tipos de pruebas que es la paramétrica y no paramétrica, el término "prueba de Quandt Goldfeld" significa que es la prueba paramétrica (Stephanie 2016).

$$F(n_2 - k, n_1 - k) = \frac{\frac{RSS_2}{n_2} - k}{\frac{RSS_1}{n_1} - k}$$

donde n_1 es el tamaño del grupo 1, n_2 es el tamaño del grupo 2 y k son los grados de libertad.

En el software R se utiliza la función *gqtest()* que arroja resultados como el estadístico de Quandt Goldfeld y su probabilidad, ese último valor se compara con el nivel de significancia α . Si la probabilidad es menor que el nivel de significancia se rechaza la hipótesis nula de Homogeneidad.

2.4. Machine Learning

Machine Learning o aprendizaje automático es un ámbito de la inteligencia artificial que nació en el año de 1960 por la combinación de ciencias de la computación y las neurociencias. Su funcionamiento consiste en aprender patrones de un conjunto extenso de datos para realizar una función específica. Lo que aprende no es la máquina, sino un algoritmo el cual describe o da conclusiones del comportamiento de los datos. Cabe, mencionar que un algoritmo es una secuencia de instrucciones que da solución a un determinado problema.

El Machine Learning es la ciencia de programación cuyo objetivo es que los ordenadores puedan aprender de los datos. De una manera más técnica, es el campo de estudio que da a los ordenadores la habilidad de aprender sin haber sido programados. Por ejemplo, el filtro de los correos spam es un programa de aprendizaje automático que puede aprender a reconocer mensajes spam dándole ejemplo de emails que son spam (Cabezón Manchado 2018).

Machine Learning nos ayuda en muchos ámbitos:

- Problemas en los que existe solución, pero es muy difícil su mantenimiento o problemas con muchas reglas que dificultan su programación. El Machine Learning simplifica los códigos y obtiene mejores resultados.
- Problemas complejos de los que no existe buena solución. Las nuevas técnicas pueden encontrar una solución.
- Problemas cuyo entorno va cambiando. El sistema de Machine Learning se adapta a los nuevos datos.
- Machine Learning puede recoger información sobre problemas muy complejos y con una gran cantidad de datos.

(Cabezón Manchado 2018)

Algunos ejemplos de aprendizaje automático son:

- Predecir la probabilidad de que un cliente regrese dentro de los 30 días.
- Predecir el tráfico urbano.
- Predecir la demanda eléctrica del siguiente mes.
- Predecir la temperatura promedio de los 10 días siguientes.
- Etc.

Los algoritmos del aprendizaje automático se dividen en aprendizaje supervisado, aprendizaje no supervisado, aprendizaje semisupervisado y aprendizaje por refuerzo.

2.4.1. *Aprendizaje Supervisado*

El aprendizaje supervisado consiste en la predicción de un resultado dado otras variables. El algoritmo que se utiliza para este caso, modela las relaciones existentes entre la variable que se va a predecir y las variables predictoras. Según Kuhn y Johnson (2013) el modelo predictivo consiste en el procesamiento y división de los datos, y el desarrollo de un modelo matemático que elabora una predicción precisa.

Este tipo de aprendizaje es el más utilizado y es aquél en el que el conjunto de entrenamiento que le damos a la máquina incluye las soluciones. Estas soluciones se denominan etiquetas. Las redes neuronales recurrentes pertenecen a este tipo de entrenamiento. Otros ejemplos de este tipo son: regresión lineal, el método de k-vecinos más cercanos, regresión logística, árboles de decisión y random forests, etc. (Cabezón Manchado 2018)

El aprendizaje supervisado se refiere a un modelo específico de Machine Learning en el cual el proceso de generación de conocimiento se realiza con un grupo de ejemplos o datos etiquetados en

los que los resultados que arroja la operación son conocidos previamente. Este tipo de modelo aprende de estos resultados e incorpora ajustes en los parámetros interiores para poder adaptarse a datos nuevos que ingresan al sistema.

La predicción realizada con datos cuantitativos se le conoce como regresión y con datos cualitativos como clasificación. Los ejemplos más comunes realizado con este aprendizaje son: reconocimiento de voz, traducción de palabras, predicción del precio de una casa, predicción de la temperatura, etc.

2.4.2. *Aprendizaje No Supervisado*

A menudo el aprendizaje no supervisado se realiza como parte de un análisis exploratorio de datos. Sin embargo, el ejercicio tiende a ser más subjetivo y no existe un objetivo simple para el análisis, como la predicción de una respuesta. Además, puede ser difícil evaluar la calidad de los resultados obtenidos de métodos de aprendizaje no supervisados (Boehmke y Greenwell 2019).

El aprendizaje no supervisado es aquél en el que el conjunto de entrenamiento no incluye las soluciones, es decir, está desetiquetado por lo que el sistema intenta aprender sin profesor (Cabezón Manchado 2018).

En el aprendizaje no supervisado se incluyen conjuntos de datos sin etiquetar en los que no se conoce previamente la estructura que estos poseen. En este tipo de aprendizaje se busca obtener información clave o importante sin conocer previamente la referencia de variables de salida, explorando la estructura de los datos que no están etiquetados. Dentro de este tipo de aprendizaje hay dos categorías específicas que se conocen como clustering y reducción de la dimensionalidad. El clustering es una técnica exploratoria para analizar datos en la que se organiza la información por grupos desconociendo de forma previa la estructura que los compone. Esto se hace con la finalidad de obtener grupos de datos con características similares. En el caso de la reducción de la dimensionalidad se utiliza datos de alta complejidad que demandan mayor capacidad de almacenamiento. Esta técnica funciona determinando correlaciones entre las características que se presentan en el conjunto de datos, disminuyendo las redundancias de información y reduciendo el tiempo de análisis para poder obtener de forma más eficiente la información considerada de mayor valor.

2.4.3. *Aprendizaje Semisupervisado*

El aprendizaje semisupervisado es una combinación de los aprendizajes anteriores en donde parte de los datos están etiquetados (con respuesta) mientras que otros no tienen la solución. Algunos ejemplos son redes como Red de Creencia Profunda o en inglés *Deep belief networks* (DBN), etc. (Cabezón Manchado 2018)

En el aprendizaje semisupervisado se emplean pocos datos etiquetados y muchos datos no etiquetados como parte del conjunto de entrenamiento. Este aprendizaje trata de explorar la información estructural que contienen los datos no etiquetados con el objetivo de generar modelos predictivos que funcionen mejor que los que sólo utilizan etiquetados. Un modelo sobre el uso de datos no etiquetados para construir un modelo de clasificación es el método self-learning o decision-directed learning. La primera fase de dicho algoritmo consiste en aprender un clasificador con los pocos datos etiquetados. Posteriormente, el clasificador es usado para predecir los datos no etiquetados y sus predicciones de mayor fiabilidad son añadidas al conjunto de entrenamiento. Finalmente, el clasificador es reentrenado con el nuevo conjunto de entrenamiento (Ibáñez y Rubén 2019).

2.4.4. Aprendizaje Por Refuerzo

En este tipo de aprendizaje la base es el refuerzo, es decir, el algoritmo tiene la capacidad de aprender de acuerdo a pruebas y errores. A priori se conocen los resultados, pero no se sabe cuáles son las decisiones correctas para llegar a obtenerlos, el proceso del algoritmo es asociar los patrones de éxito y repetirlos hasta perfeccionarlos.

El aprendizaje reforzado tiene como finalidad construir modelos que aumenten el rendimiento tomando como base el resultado o recompensa que se genera por cada interacción realizada. Esta recompensa es el producto de una acción correcta o conjunta de datos devueltos que entran en una medida específica.

El modelo, mediante un agente, utiliza la recompensa como parámetro de ajuste en su comportamiento para acciones futuras, de forma tal de que la acción nueva cumpla igualmente con el objetivo o acción correcta y así obtener una recompensa máxima (Graph).

En este trabajo de investigación nos centraremos en el aprendizaje automático supervisado de las redes neuronales.

2.5. Redes Neuronales

Neuronas biológicas y Neuronas artificiales

El cerebro consiste en un conjunto de células cerebrales (neuronas) conectadas entre sí. Una neurona toma los impulsos eléctricos como señales, procesan el mensaje y lo envían a otras neuronas.

Una neurona consta de cuatro partes:

- **Dendritas.** – Aceptan los impulsos eléctricos de entrada.
- **Cuerpo.** – Es la parte donde se procesa la información.

- **Axón.** - Transforma la información de entrada de tal forma que esta información pueda ser recibida por la siguiente neurona.
- **Sinapsis.** – Es el contacto electroquímico entre neuronas.

En la figura 2-2 se puede observar las partes de una neurona biológica y una neurona artificial.

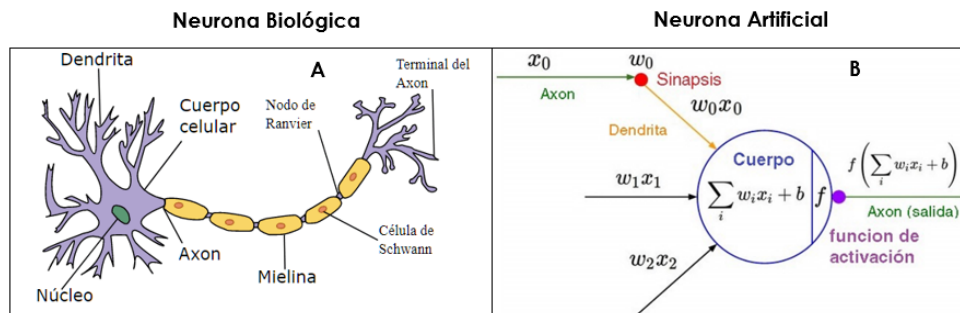


Figura 2-2. Neurona biológica y Neurona artificial

Fuente: (Khepri 2018)

Las neuronas artificiales que se interconectan en diversas redes son las encargadas del aprendizaje. Cada neurona recibe y combina señales desde otras neuronas. Mediante la sinapsis es posible la transmisión de información entre estas neuronas a través de las dendritas. Si la señal combinada es lo suficientemente fuerte, el nervio libera neurotransmisores. Según el tipo de neurotransmisor, las neuronas pueden excitarse si reciben el estímulo o inhibirse si no llega, generando una respuesta u otra según el caso (Cabezón Manchado 2018). En la figura 2-2 B se puede observar una representación de la neurona artificial.

En una neurona artificial las entradas son variable inputs que nos sirven como variables explicativas. Estas variables tienen que llevar asociados unos pesos iniciales que pueden ser aleatorios o fijados. Por tanto, los pesos sinápticos w_{ij} que representa la fuerza de una conexión sináptica entre la neurona presináptica i y la postsináptica j . Una neurona puede recibir múltiples entradas simultáneamente y cada entrada tiene su propio peso sináptico el cual proporciona la importancia de la entrada dentro de la función de activación de la neurona. Estos pesos representan un estado de la memoria, ya que si un peso es cercano a cero se considera que no existe conexión entre esas neuronas. Estos coeficientes pueden adaptarse dentro de la red y van siendo modificados en respuesta de los ejemplos de muestreo de acuerdo a las reglas de entrenamiento (Cabezón Manchado 2018).

Redes Neuronales Artificiales o RNAs

Las redes neuronales artificiales (RNA) están inspiradas en el funcionamiento del cerebro humano, aunque no llegan a tal capacidad, estas son muy útiles a la hora de realizar diferentes actividades como el reconocimiento de voz, de escritura, procesamiento de video entre otras. Su finalidad es

identificar patrones o modelos mediante técnicas algorítmicas convencionales para solucionar diferentes tipos de problemas.

Las RNAs son modelos computacionales que surgieron como un intento de conseguir formalizaciones matemáticas acerca de la estructura del cerebro, imitan la estructura hardware del sistema nervioso, centrándose en el funcionamiento del cerebro humano, basado en el aprendizaje a través de la experiencia, con la consiguiente extracción de conocimiento a partir de la misma. (Flórez y Fernández 2008)

Las RNAs están formadas por una serie de neuronas artificiales, que constituyen dispositivos simples de cálculo que, bien a partir de un vector de entrada procedente del mundo exterior o a partir de estímulos recibidos de otras neuronas, proporcionan una respuesta única. Están estructuradas por tres tipos de neuronas , como lo enuncia en (Flórez y Fernández 2008):

Neuronas de entrada: Se encuentran ubicadas en la capa de entrada y reciben señales desde el entorno, provenientes de sensores o de otros sectores del sistema.

Neuronas de salida: Se encuentran en la capa de salida y envían una señal fuera del sistema una vez finalizado el tratamiento de la información.

Neuronas ocultas: Se encuentran en las capas ocultas y reciben estímulos y emiten salidas dentro del sistema, sin tener contacto con el exterior del sistema.

A continuación, se presenta una estructura básica de una red neuronal artificial.

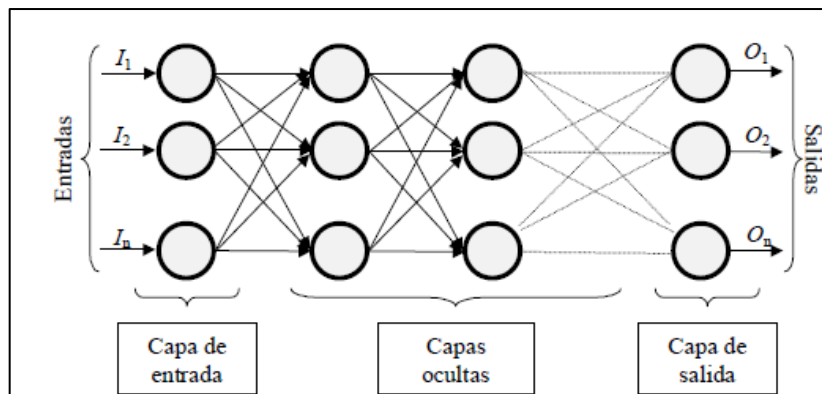


Figura 3-2. Estructura básica de una red neuronal artificial

Fuente: Match, D. 2001

2.5.1. Elementos básicos

2.5.1.1. Parámetros

Para el entrenamiento de la red se requieren de los siguientes elementos:

Número de capas: En una red neuronal se debe definir el número de capas con las que se va a trabajar, como mínimo debe tener la capa de entrada y salida. Las capas intermedias no son obligatorias y sirven para reflejar relaciones no lineales entre las variables.

Número de neuronas: Además, se debe definir el número de neuronas, en la capa de entrada el número de neuronas depende de los inputs o el número de variables independientes, y en la capa de salida depende de los outputs o el número de predicciones. El número de neuronas en la/las capa/capas ocultas va a depender de la cantidad de error que arroja la red, para ello se debe realizar pruebas para elegir el adecuado para el tipo de datos que se va a trabajar. Cabe mencionar que no existe una regla determinada que indique el número de capas y el número de neuronas que debe tener una red neuronal artificial.

Epochs: Indica el número de veces que se va a repetir los datos de entrenamiento en la red. Cuanto mayor sea el número de epochs se irá disminuyendo el error de ajuste.

Batch_size: Indica el tamaño de las sub-muestras generadas para calcular el vector gradiente y optimizar el modelo.

Función de coste: La función de coste nos indicará cómo de malo es nuestro modelo en términos de su habilidad para estimar la relación entre las variables de entrada x y el target y . El objetivo de la red será disminuir o minimizar esta función. Algunos ejemplos son: MSE (mean squared error), MAE (mean absolute error), MAPE (mean absolute percentage error) para variables continuas, y para variables binarias existen binary cross entropy, categorical crossentropy, etc.

Optimizador (optimizer): Es el mecanismo a través del cual la red se actualiza, basándose en los datos que ve y en la función de coste. Entre estas funciones se encuentra, el gradiente descendente y la función Adam.

2.5.1.2. Tipos de funciones

Función de entrada (input function)

La neurona trata a muchos valores de entrada como si fuera uno solo, esto recibe el nombre de entrada global. Por lo tanto, lo que se quiere es combinar las entradas (x_{i1}, x_{i2}, \dots) dentro de la entrada global, x_{ij} indica que la variable explicativa x_j entra a la neurona i . Esto se logra a través de la función de entrada, la cual se calcula a partir del vector de entrada. La función de entrada puede describirse como sigue:

$$input_i = (x_{i1} w_{i1}) * (x_{i2} w_{i2}) * \dots * (x_{in} w_{in})$$

donde $input_i$ es la entrada a la neurona i , $*$ representa al operador apropiado que puede ser la suma, el producto, el máximo, etc., n es el número de entradas a la neurona N_i y w_i es el peso.

Los pesos que generalmente no están restringidos cambian la medida de influencia que tienen los valores de entrada, es decir, que permiten que un gran valor de entrada tenga solamente una pequeña influencia si estos son lo suficientemente pequeños.

(Matich 2001)

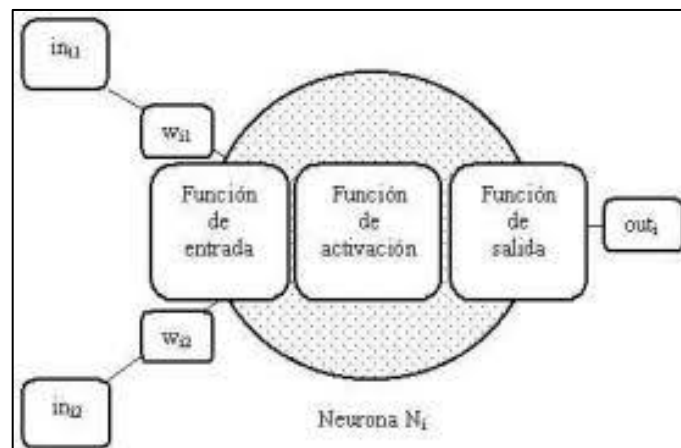


Figura 4-2. Neurona con dos entradas y una salida

Fuente: Matich, D. 2001

Función de activación

Una neurona artificial tiene un estado de activación que puede ser activa (excitada) o inactiva (no excitada), la función de activación determina este estado transformando la entrada global en un valor que usualmente está dado en un intervalo de (0 o 1) o de (-1 a 1). Cuando una neurona está totalmente inactiva toma valores de (0 o -1) y cuando esta activa toma el valor de (1).

La función de activación f_i calcula el estado de actividad de una neurona. Determina el estado de activación actual $a_i(t)$ de la neurona en base al potencial resultante h_i y al estado de activación anterior de la neurona $a_i(t - 1)$. El estado de activación de una neurona i para un determinado instante de tiempo t puede ser expresado de la siguiente manera:

$$a_i(t) = f_i(a_i(t - 1), h_i(t))$$

En la mayoría de los casos se suele ignorar el estado de activación anterior de la neurona quedando de la siguiente forma:

$$a_i(t) = f_i(h_i(t))$$

(Cabezón Manchado 2018)

Las funciones de activación más utilizadas son:

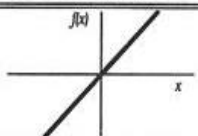
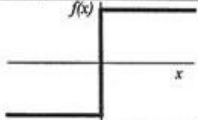
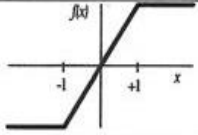

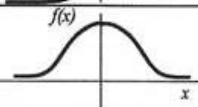
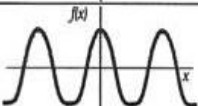
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura 5-2. Funciones de activación habituales

Fuente: <http://grupo.us.es/gtocom/pid/pid10/RedesNeuronales.htm>

Función de salida (output function)

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona i (out_i); por ende, la función de salida determina que valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida pasa a la neurona subsiguiente. Normalmente, no cualquier valor es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$. También pueden ser binarios $\{0, 1\}$ o $\{-1, 1\}$.

Dos de las funciones de salida más comunes son:

- Identidad. - Este es de tipo de función más sencillo, tal que la salida es la misma que la entrada.
- Binaria: $\begin{cases} 1 & \text{si } a_i(t) \geq \varepsilon_i \\ 0 & \text{de lo contrario} \end{cases}$; donde ε_i es el umbral de aceptación

2.5.1.3. Aprendizaje de una red neuronal

Las redes neuronales manejan dos tipos de información. La primera, es la información volátil, que se refiere a los datos que están usando y varían con la dinámica de la red. Esta información se encuentra almacenada en el estado dinámico de las neuronas. Es segundo tipo es la información no

volátil que se mantiene para recordar los patrones aprendidos y se encuentran almacenadas en los pesos sinápticos.

El proceso de aprendizaje o entrenamiento de una red neuronal consiste en encontrar la relación entre los pesos que permiten desarrollar la tarea para que la red ha sido diseñada, como puede ser predecir o clasificar.

El aprendizaje se basa en el entrenamiento de la red con patrones, que usualmente son llamados patrones de muestra o entrenamiento. El proceso usual del algoritmo es que la red ejecuta los patrones iterativamente, cambiando los pesos de la sinapsis, hasta que convergen a un conjunto de pesos óptimos que representan a los patrones lo suficientemente bien. Esto es, sus pesos sinápticos se ajustan para dar respuestas correctas al conjunto de patrones de entrenamiento que le hemos mostrado.

(Cabezón Manchado 2018)

Para el aprendizaje de las RNAs son necesarias técnicas de minimización del error o la optimización de alguna función de recompensa para modificar el valor de los pesos sinápticos en función de las entradas disponibles y con ello optimizar la respuesta de la red a las salidas que deseamos. El objetivo del proceso de entrenamiento es encontrar los pesos que minimicen la función del error escogida. Las etapas en el proceso de aprendizaje son:

- **Inicialización de la red.** - El comienzo del aprendizaje, en $n = 0$ se generan unos pesos aleatorios para todas las conexiones. Con estos pesos se puede calcular un primer error de predicción o clasificación.
- **Hacia adelante.** – La información pasa desde la entrada hasta la salida en una única dirección a través de las funciones de activación y de los pesos. Para cada instante de tiempo n se realiza un nuevo patrón de entrada y se intentan refinar los pesos buscando mejorar el nivel de error de la etapa anterior.
- **Asignación del error.** - El algoritmo se detiene cuando el error obtenido es menor que una cota preestablecida, cuando el error no decrece sencillamente o cuando se ha alcanzado el número de iteraciones. En caso contrario se pasa a la siguiente etapa.
- **Propagación del error.** – El error obtenido en la capa de salida es propagado hacia atrás para modificar los pesos.
- **Ajuste.** – Se ajustan los pesos usando el método Adam o el método del gradiente descendente buscando la minimización del error.

(Cabezón Manchado 2018)

2.5.2. *Tipos de redes neuronales*

Las redes neuronales artificiales se pueden clasificar por su topología es decir de acuerdo al número de capas pueden ser estas de entrada, salida u ocultas y de acuerdo a la direccionalidad de las conexiones de las neuronas. De tal forma las redes se clasifican en:

Redes Monocapa: Estas redes son aquellas que poseen una sola capa. Para unirse se crean conexiones entre las neuronas de la capa. Entre las redes monocapa existen algunas en las que las neuronas pueden estar conectadas consigo mismas y se denominan autorecurrentes.

Redes Multicapa: Estas redes poseen múltiples capas de neuronas. Además, se pueden clasificar de acuerdo a la manera que se conectan las capas.

- Feedforward: También conocidas como estáticas estas redes tienen un único sentido (hacia delante) en las conexiones de las capas desde los inputs hasta los outputs. Entre estas redes se encuentran el perceptrón, adaline, etc.
- Feedback: Conocidas también como dinámicas, estas redes tienen conexiones entre capas tanto hacia delante como hacia atrás y por tanto la información puede volver a capas anteriores en la dinámica de la red.

(Cabezón Manchado 2018)

2.6. **Redes neuronales recurrentes**

Las redes neuronales recurrentes (RNN) son redes con bucles que hace que la información quede guardada como una especie de memoria. Estas redes no tienen una estructura de capas definida por lo que las neuronas se conectan arbitrariamente unas a otras o en sí mismas. Son útiles para la predicción de secuencias, análisis de textos, sonidos y videos.

Las redes neuronales recurrentes tienen una gran habilidad para aprender problemas difíciles que tratan sobre datos de series temporales. Estas redes tienen estados ocultos distribuidos en el tiempo que permiten almacenar mucha información sobre el pasado de forma eficiente.

El funcionamiento de estas redes es como cuando una persona lee un texto, ella conoce y entiende las palabras que ve por qué lo tiene guardado el significado de sus palabras en la memoria, así mismo las RNN guardan información de eventos pasados.

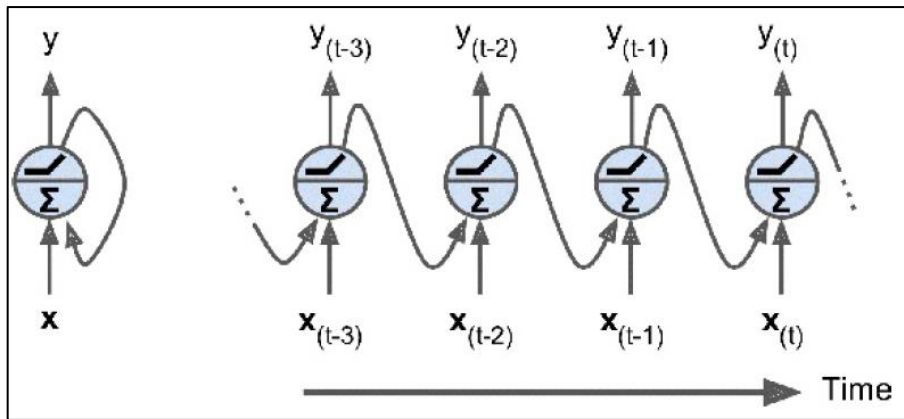


Figura 6-2. Neurona recurrente (izquierda), desarrollo en el tiempo (derecha)

Fuente: (Géron 2017)

En las RNN las salidas de las neuronas dependen de los actuales valores de las variables de entrada a la red, de los anteriores valores de entrada, de los anteriores valores de salida, etc.

En la figura 6-2 vemos que en cada paso temporal t , esta neurona recibe las $x_{(t)}$ como la salida del paso anterior, $y_{(t-1)}$. Si representamos este proceso en el eje x como si fuera el tiempo, obtendríamos la parte derecha de la figura 6-2.

Con esto podemos formar fácilmente una capa con neuronas recurrentes. En cada tiempo t todas las neuronas reciben el vector de entrada $x_{(t)}$ y el vector de salida del paso anterior $y_{(t-1)}$ como se muestra a continuación:

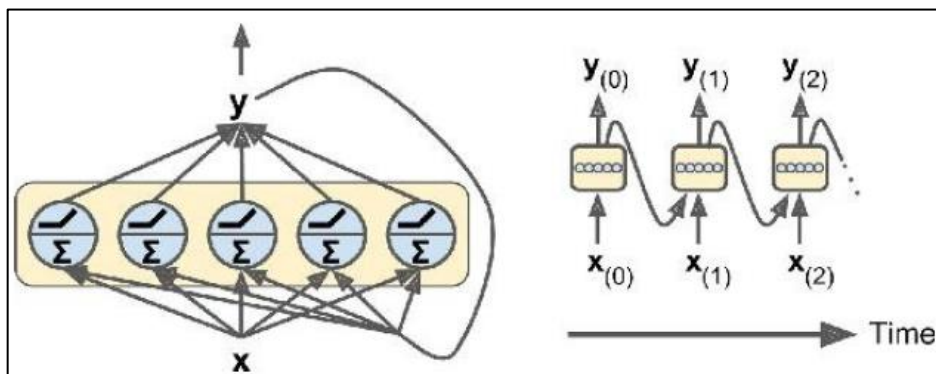


Figura 7-2. Capa de neuronas recurrentes (izquierda), desarrollo en el tiempo (derecha).

Fuente: (Géron 2017)

Cada neurona recurrente tiene dos conjuntos de pesos; uno para los inputs $x_{(t)}$ y otro para las salidas del paso temporal anterior $y_{(t-1)}$. Llamemos a estos pesos w_x y w_y . La salida $y_{(t)}$ de una neurona recurrente se puede calcular de la siguiente manera:

$$y_{(t)} = \Phi(x_{(t)}^T \cdot w_x + y_{(t-1)}^T \cdot w_y + b)$$

Donde b es el sesgo (bias) y \emptyset es la función de activación. Podemos resumir la expresión mediante la expresión anterior mediante el álgebra lineal de la siguiente manera:

$$Y_{(t)} = \emptyset(X_{(t)} \cdot W_x + Y_{(t-1)} \cdot W_y + b) = \emptyset([X_{(t)} \ Y_{(t-1)}] \cdot W + b)$$

siendo $W = [W_x \ W_y]^T$ y donde:

- $Y_{(t)}$ es una matriz $m \times n_{inputs}$ que contiene los outputs del marco temporal t para cada *batch* (conjunto de datos que entra a la neurona) en este caso $m \times n_{inputs}$ es el número de neuronas de la capa.
- $X_{(t)}$ es una matriz $m \times n_{inputs}$ que contiene las variables inputs de las m observaciones y n_{inputs} es el número de variables explicativas de entrada.
- W_x es una matriz $n_{inputs} \times n_{neuronas}$ conteniendo los pesos de las conexiones para las entradas del paso temporal actual.
- W_y es una matriz $n_{neuronas} \times n_{neuronas}$ conteniendo los pesos de las conexiones para las salidas del paso temporal anterior.
- Las matrices de pesos W_x y W_y se pueden juntar en la matriz W que será $(n_{inputs} + n_{neuronas}) \times n_{neuronas}$
- b es un vector de 1's de tamaño $n_{neuronas}$ que permite añadir un término constante (bias) al modelo.

Hay que darse cuenta de que $Y_{(t)}$ es una función de $X_{(t)}$ y de $Y_{(t-1)}$, quien a su vez es función de $X_{(t-1)}$ y de $Y_{(t-2)}$, quién es función de $X_{(t-2)}$ y de $Y_{(t-3)}$, etc. Esto hace que $Y_{(t)}$ sea función de todas las variables de entrada desde el tiempo inicial $t = 0$ (esto es, de $X_{(0)}, X_{(1)}, \dots, X_{(t)}$). En $t = 0$ no hay outputs anteriores luego se asumen como ceros.

(Cabezón Manchado 2018)

2.6.1. Red de Elman

La red de Elman es la más simple y su estructura es la base de las demás redes neuronales recurrentes, su estructura consiste en una capa de entrada, una capa oculta y una de salida, se le añade además una capa con unidades de contexto también llamadas unidades de retardo, el número de neuronas de esta capa debe ser igual al número de neuronas de la capa oculta. La red tiene en su arquitectura la retroalimentación que permite crear la temporalidad, consiguiendo que la red posea memoria, el tipo de memoria es a corto plazo.

El proceso de memoria se produce a través de las unidades de retardo que son alimentadas por las neuronas de la capa oculta. Los pesos de las conexiones entre la capa oculta y las unidades de retardo

son fijos e iguales a 1 lo que permite que siempre se tenga una copia de los valores de salida de la capa oculta del paso previo (Cabezón Manchado 2018).

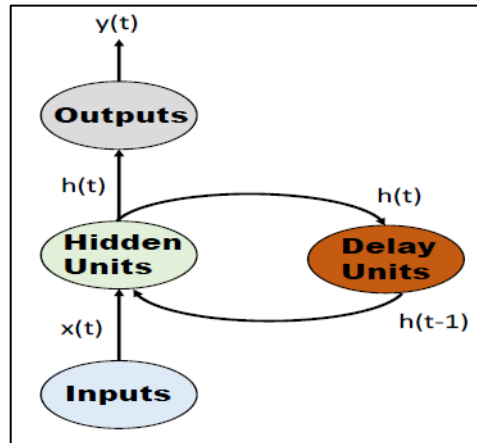


Figura 8-2. Red Neuronal de Elman

Fuente: (Lewts 2016)

En la figura 8-2 vemos el esquema generalizado de la red de Elman, observando que la salida de la capa oculta se almacena en la capa de retardo, y que luego esta sirve de entrada en el siguiente paso para la capa oculta.

La ecuación matemática que define el proceso es:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h),$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

donde x_t es el vector con las variables de entrada, h_t es el vector de la capa oculta, y_t es el vector de salida, $W = [W_x \ W_y]^T$ y U_h son las matrices de pesos y $b = [b_h \ b_y]^T$ el bias. σ_h y σ_y son las funciones de activación de las respectivas capas.

(Cabezón Manchado 2018)

2.6.2. Red de Jordan

La arquitectura de una red de Jordan es similar al de la red de Elman (capa de entrada, de salida, oculta y de contexto). La diferencia radica en que las entradas a las neuronas de la capa de contexto se realizan desde la capa de salida. En la figura 9-2 vemos la estructura generalizada de una red de Jordan.

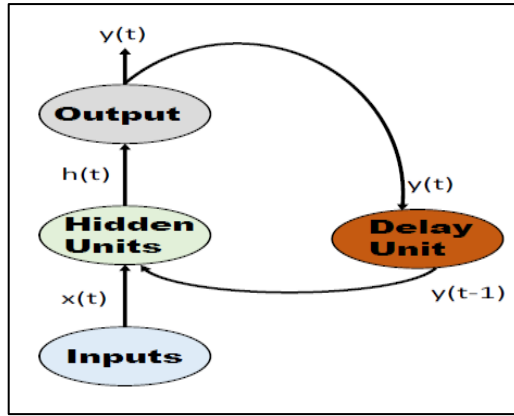


Figura 9-2. Red Neuronal de Jordan

Fuente: (Lewts 2016)

La ecuación matemática que define el proceso es:

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h),$$

$$y_t = \sigma_y(W_y h_t + b_y)$$

donde x_t es el vector con las variables de entrada, h_t es el vector de la capa oculta, y_t es el vector de salida, $W = [W_x \ W_y]^T$ y U_h son las matrices de pesos y $b = [b_h \ b_y]^T$ el bias. σ_h y σ_y son las funciones de activación de las respectivas capas.

(Cabezón Manchado 2018)

2.6.3. Red LSTM

Las redes de memoria a corto y largo plazo también conocidas como LSTM, son capaces de recordar un dato relevante de la secuencia y preservarlo por varios instantes de tiempo por tanto puede tener una memoria a corto plazo como las redes recurrentes básicas y memoria a largo plazo, funciona de forma similar a como nuestro cerebro analiza las secuencias. Es parecida a las redes de Elman y Jordan sólo que sustituye las neuronas y las unidades ocultas por un bloque de memoria, este bloque es una célula LSTM. En la siguiente imagen vemos como es una célula LSTM por dentro:

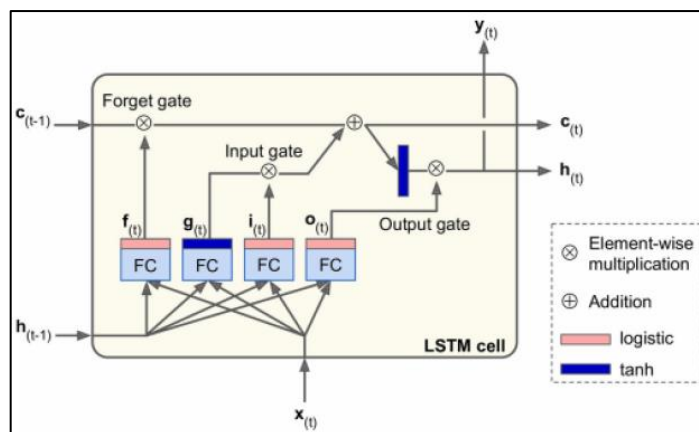


Figura 10-2. Red Neuronal LSTM

Fuente: Géron, Aurélien. 2017

Si no nos fijamos en el interior de la caja, una célula LSTM es igual a una célula normal salvo que su estado se divide en dos vectores, $h_{(t)}$ y $c_{(t)}$ (c viene de célula). $h_{(t)}$ hace referencia al estado de corto plazo y $c_{(t)}$ al de largo plazo.

La principal idea de la célula es que la red pueda aprender qué almacenar en el estado de largo plazo, qué desechar y qué leer de ellos. Si nos fijamos en el término de largo plazo $c_{(t-1)}$ atraviesa la red de izquierda a derecha, primero pasa por una puerta de olvido (*forget gate*), deshaciéndose de algunos datos almacenados, luego añade nuevas informaciones que llegan de la puerta de entrada (*input gate*) y el resultado $c_{(t)}$ se envía directamente fuera de la célula y se deja como está y servirá de entrada para la siguiente época. Por lo tanto, en cada paso algunos recuerdos son eliminados mientras que otros nuevos se añaden. Además, una vez añadidos nuevos recuerdos, se hace una copia del estado a largo plazo y pasa por la función tangente hiperbólica para después ser filtrado por la puerta de salida (*output gate*). Este paso produce el estado a corto plazo $h_{(t)}$ (que es igual a la salida de la célula $y_{(t)}$ para este paso temporal t).

Ahora vamos a ver de dónde vienen los nuevos recuerdos que se memorizan y cómo trabajan estas puertas.

Primero se observa que el vector de entrada actual $x_{(t)}$ y el estado de corto plazo anterior $h_{(t-1)}$ van a parar a cuatro capas conectadas. Todas ellas con un distinto objetivo:

- La capa principal es la que tiene como salida $g_{(t)}$. Tiene el típico rol de analizar las entradas actuales $x_{(t)}$ y el estado a corto plazo anterior $h_{(t-1)}$. En una célula normal sólo habría esta capa y su salida iría directamente a $y_{(t)}$ y $h_{(t)}$, sin embargo, en una célula LSTM la salida de esta capa no va directamente fuera, se almacena parcialmente el estado a largo plazo.

- Las otras tres capas son controladores de puertas (*gate controllers*) ya que usan la función logística como función de activación que varía entre 0 y 1. Como se ve en la figura 10-2 sus salidas van directas a operaciones multiplicativas, por lo que si la salida es cercana a 0 menos información pasa y se cierra la puerta mientras que si se acerca a 1 más información fluye y se abre la puerta. Más específicamente:
 - La *forget gate* ($f_{(t)}$) controla qué partes del estado a largo plazo deben ser borradas.
 - La *input gate* ($i_{(t)}$) controla qué partes de $g_{(t)}$ deben ser añadidas al estado de largo plazo.
 - La *output gate* ($o_{(t)}$) controla qué partes del estado de largo plazo deben ser leídas y su salida es tanto $h_{(t)}$ como $y_{(t)}$.

En resumen, una célula LSTM puede aprender a reconocer datos inputs importantes (que es rol de la puerta de entrada), almacenarla en el estado de largo plazo, aprender a guardarla siempre que sea necesario (cuyo rol es el de la puerta de olvido) y aprender a extraer su información siempre y cuando se necesite. Todo esto es la razón de por qué LSTM ha tenido tanto éxito a la hora de encontrar patrones a largo plazo en series temporales, largos textos, grabaciones de audio, etc.

Las siguientes ecuaciones resumen cómo calcular el estado de largo plazo, el de corto plazo y su salida para un paso temporal.

$$\begin{aligned}
 i_{(t)} &= \sigma(W_{xi}^T \cdot x_{(t)} + W_{hi}^T \cdot h_{(t-1)} + b_i) \\
 f_{(t)} &= \sigma(W_{xf}^T \cdot x_{(t)} + W_{hf}^T \cdot h_{(t-1)} + b_f) \\
 o_{(t)} &= \sigma(W_{xo}^T \cdot x_{(t)} + W_{ho}^T \cdot h_{(t-1)} + b_o) \\
 g_{(t)} &= \tanh(W_{xg}^T \cdot x_{(t)} + W_{hg}^T \cdot h_{(t-1)} + b_g) \\
 c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \\
 y_{(t)} = h_{(t)} &= o_{(t)} \otimes \tanh(c_{(t)})
 \end{aligned}$$

Donde;

- $W_{xi}, W_{xf}, W_{xo}, W_{xg}$ son las matrices de pesos de las conexiones entre las cuatro capas y el vector de entrada $x_{(i)}$.
- $W_{hi}, W_{hf}, W_{ho}, W_{hg}$ son las matrices de pesos de las conexiones entre las cuatro capas y el estado de corto plazo previo $h_{(t-1)}$.
- b_i, b_f, b_o, b_g son términos de bias para las cuatro capas.

(Cabezón Manchado 2018; Géron 2017)

2.6.4. Red GRU

Las células GRU (Gated Recurrent Unit, unidades recurrentes cerradas) son una versión simplificada de las células LSTM. A continuación, se observa se estructura:

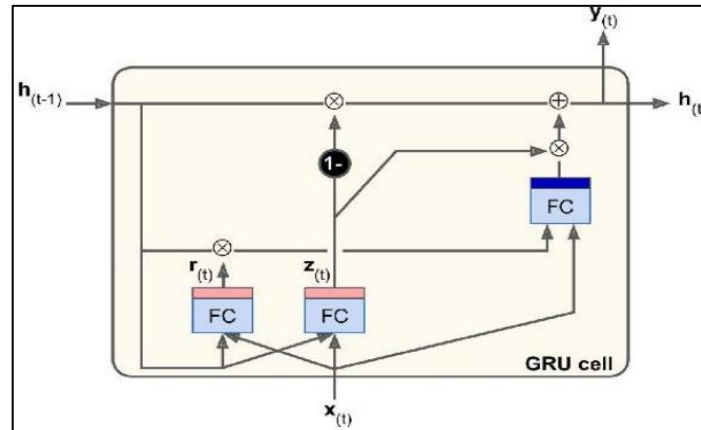


Figura 11-2. Red Neuronal GRU

Fuente: Géron, Aurélien. 2017

Las principales simplificaciones son las siguientes:

- Los dos vectores de estado (corto y largo plazo) se juntan ahora en un único vector $h_{(t)}$.
- Ahora sólo hay un controlador de las puertas que controla tanto la puerta de olvido como la puerta de entrada. Si la puerta de control vale 1, la puerta de entrada se abre y la de olvido se cierra y si la de control vale 0 ocurre lo contrario.
- No hay puerta de salida, la salida siempre es el vector de los estados entero. Sin embargo, hay una nueva capa de control que controla qué partes del estado anterior deben ser mostradas en la capa principal.

Las siguientes ecuaciones resumen lo anterior explicado:

$$z_t = \sigma(W_{xz}^T * x_{(t)} + W_{hz}^T * h_{(t-1)})$$

$$r_t = \sigma(W_{xr}^T * x_{(t)} + W_{hr}^T * h_{(t-1)})$$

$$g_t = \tanh(W_{xg}^T * x_{(t)} + W_{hg}^T * (r_{(t)} \otimes h_{(t-1)}))$$

$$h_t = (1 - z_{(t)}) \otimes \tanh(W_{xg}^T * h_{(t-1)} + z_{(t)} \otimes g_{(t)})$$

(Cabezón Manchado 2018; Géron 2017)

2.7. Medidas de evaluación de la precisión del pronóstico

Para identificar si un modelo obtenido es adecuado para explicar el comportamiento de los datos se debe realizar una evaluación de precisión del pronóstico debido a que los residuos obtenidos mediante el ajuste del modelo y los datos reales no es una indicación confiable de cuan grandes pueden ser los errores de pronóstico reales. Se debe tener en cuenta los siguientes puntos con respecto al ajuste del modelo.

- Un modelo que se ajuste bien a los datos de entrenamiento no necesariamente pronosticará bien.
- Siempre se puede obtener un ajuste perfecto utilizando un modelo con suficientes parámetros.
- El ajuste excesivo de un modelo a los datos es tan malo como no identificar un patrón sistemático en los datos.

(Hyndman y Athanasopoulos 2018)

Comúnmente al conjunto de datos se le divide en dos partes que es la parte de entrenamiento y prueba. La primera parte se utiliza para estimar un parámetro de una técnica de predicción y la segunda se usa para validar el modelo.

2.7.1. Errores de pronóstico

El error de pronóstico se obtiene por la diferencia de los datos reales (y_t) con los datos pronosticados (\hat{y}_t). Estos errores se clasifican de acuerdo a la dependencia de escala y en porcentajes de error.

2.7.1.1. Errores dependientes de la escala

Estos errores están en la misma escala que los datos por lo tanto no se puede hacer comparaciones entre series que tengan diferente tipo de escala.

2.7.1.1.1. Mean Square Error (MSE)

El Error Cuadrático Medio se obtiene mediante el cociente entre la sumatoria del error al cuadrado y el número de datos pronosticados.

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

Las unidades del MSE no son las mismas de los datos originales ni del estimador debido a que se eleva al cuadrado el error. Además, al elevar al cuadrado las diferencias hace que los datos negativos y positivos no se cancelen entre sí y otorga mayor peso a aquellos errores de mayor tamaño siendo útil en los casos en los que el pronóstico implica valoración de los errores. Además, el error es sensible a datos atípicos (Vélez Correa y Nieto Figueroa 2016).

2.7.1.1.2. Root Mean Square Error (RMSE)

El RMSE se obtiene al aplicar la raíz cuadrada al error cuadrático medio.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

Este tipo de error tiene la misma escala que los valores originales y es sensible a valores atípicos.

2.7.1.1.3. Mean Absolute Error (MAE)

El Error Absoluto Medio se obtiene del cociente entre la sumatoria de los valores absolutos de la diferencia de los valores originales con los pronosticados sobre el número total de valores pronosticados. Este error es menos afectado por los valores atípicos debido a que no están elevados al cuadrado. Su fórmula se representa de la siguiente forma:

$$MAE = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|$$

2.7.1.1.4. Median Absolute Error (MdAE)

Se obtiene al aplicar la mediana de los valores absolutos de la diferencia entre los datos originales y los pronosticados.

$$MdAE = \text{Mediana}(|y_t - \hat{y}_t|); \text{ para } t = 1, 2, \dots, n$$

2.7.1.2. Porcentajes de error

Los porcentajes de error están libres de escala por lo que se puede comparar la precisión de los pronósticos con datos que no necesariamente tengan la misma escala.

2.7.1.2.1. Mean Absolute Percentage Error (MAPE)

El Error Porcentual Absoluto Medio es un indicador de precisión del pronóstico, mide el error en términos porcentuales. Ya que es un estadístico que estima la magnitud del error es fácil de interpretar y es frecuentemente utilizado.

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

2.7.1.2.2. Median Absolute Percentage Error (MdAPE)

El porcentaje de error absoluto de la mediana es similar al MAPE pero en vez de resumir los errores en un porcentaje absoluto (APE), calcula su promedio. Es decir, que toda la APE están ordenadas

desde la más pequeña a la más grande y el APE en el medio se utiliza para denotar la mediana (Vélez Correa y Nieto Figueroa 2016).

$$MdAPE = Mediana \left(\left| \frac{y_t - \hat{y}_t}{y_t} \right| \right) ; \text{ para } t = 1, 2, \dots, n$$

2.7.1.2.3. Root Mean Square Percentage Error (RMSPE)

El RMSPE viene dado solo en valores positivos y se define como la raíz cuadrada del valor medio de los errores en términos porcentuales al cuadrado.

$$RMSPE = \sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{y_t - \hat{y}_t}{y_t} \right)^2}$$

2.7.1.2.4. Root Median Square Percentage Error (RMdSPE)

Es la mediana de los errores en términos porcentuales al cuadrado.

$$RMdSPE = \sqrt{Mediana \left(\left(\frac{y_t - \hat{y}_t}{y_t} \right)^2 \right)} ; \text{ para } t = 1, 2, \dots, n$$

2.7.1.2.5. Symmetric Mean Absolute Percentage Error (sMAPE)

El error porcentual absoluto medio simétrico se obtiene del cociente entre la diferencia absoluta entre los valores originales y los pronosticados sobre la mitad de la suma del valor absoluto del dato original y el valor absoluto de dato pronosticado, este resultado se suma y se divide para el número de valores pronosticados.

$$sMAPE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{\frac{(|y_t| + |\hat{y}_t|)}{2}}$$

2.7.1.2.6. Symmetric Median Absolute Percentage Error (sMdAPE)

El sMdAPE se define como la mediana de los valores absolutos de los errores dividido entre la mitad de la suma del valor absoluto del dato original y el valor absoluto de dato pronosticado.

$$sMdAPE = Mediana \left(\frac{|y_t - \hat{y}_t|}{\frac{(|y_t| + |\hat{y}_t|)}{2}} \right) ; \text{ para } t = 1, 2, \dots, n$$

2.7.2. Coeficiente U de Theil

El coeficiente de desigualdad U de Theil se utiliza para medir la capacidad de predicción de los modelos. Se calcula a partir de las tasas de crecimiento de los porcentajes de variación de los valores de \hat{y} e y .

$$u = \frac{\sqrt{\frac{1}{n} \sum (y - \hat{y})^2}}{\sqrt{\frac{1}{n} \sum y^2 + \frac{1}{n} \sum \hat{y}^2}}$$

Su valor varía entre 0 y 1.

Si $u = 0$: No hay ningún tipo de desigualdad.

Si $u = 1$: Hay mucha desigualdad, desigualdad máxima.

(AulaMH 2012)

2.7.3. Prueba de Diebold-Mariano

La prueba de Diebold-Mariano se utiliza para comparar la precisión del pronóstico de dos técnicas de predicción.

Sean e_i y r_i los errores para los dos pronósticos, es decir,

$$e_i = y_i - f_i \quad r_i = y_i - g_i$$

y d_i se define como uno de las siguientes funciones (u otra medición similar)

$$d_i = e_i^2 - r_i^2 \quad o \quad d_i = |e_i| - |r_i|$$

La serie temporal se llama diferencial de pérdida. La primera fórmula está relacionada con la estadística del error MSE y la segunda está relacionada con MAE.

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$$

Para $n > k \geq 1$, se define

$$\gamma_k = \frac{1}{n} \sum_{i=k+1}^n (d_i - \bar{d})(d_{i-k} - \bar{d})$$

γ_k es la autocovarianza en el retraso k .

Para $h \geq 1$, se define la estadística de Diebold-Mariano de la siguiente manera:

$$DM = \frac{\bar{d}}{\sqrt{[\gamma_0 + 2 \sum_{k=1}^{h-1} \gamma_k]/n}}$$

generalmente es suficiente usar el valor $h = n^{\frac{1}{3}} + 1$

DM sigue una distribución normal estándar: $DM \sim N(0, 1)$.

El valor crítico se denota por $z_{\alpha/2}$ en el contraste bilateral y z_{α} en el contraste unilateral.

(Zaiontz)

Con la función `dm.test` se obtiene el estadístico de prueba y el valor *p* en el software R, y tiene tres alternativas para plantear la prueba de hipótesis las cuales son:

“two.sided” – Contraste bilateral

H_0 : Los dos métodos tienen la misma precisión de pronóstico

H_1 : El método 1 y el método 2 tienen diferentes niveles de precisión

En el contraste bilateral se rechaza H_0 si $|DM| > z_{\alpha/2}$ o si el *valor p* $< \alpha$.

$$\text{valor } p = P(DM < z_{\alpha/2}) + P(DM > z_{1-\alpha/2})$$

“greater” - Contraste Unilateral derecho

H_0 : Los dos métodos tienen la misma precisión de pronóstico

H_1 : El método 2 es más preciso que el método 1

En el contraste unilateral derecho se rechaza H_0 si $DM > z_{1-\alpha}$ o si el *valor p* $< \alpha$.

$$\text{valor } p = P(DM > z_{1-\alpha})$$

“less” - Contraste Unilateral izquierdo

H_0 : Los dos métodos tienen la misma precisión de pronóstico

H_1 : El método 2 es menos preciso que el método 1

En el contraste unilateral izquierdo se rechaza H_0 si $DM < z_{\alpha}$ o si el *valor p* $< \alpha$.

$$\text{valor } p = P(DM < z_{\alpha})$$

CAPÍTULO III

3. METODOLOGÍA

En el presente trabajo de investigación se realiza la predicción del tipo de cambio EUR/USD utilizando los modelos ARIMA y redes neuronales recurrentes de tipo Elman y LSTM.

La serie temporal representa la cotización diaria del tipo de cambio EUR/USD de lunes a viernes, los datos se obtuvieron de los registros realizados por la plataforma financiera Investing.com desde el 2 de enero de 1998 al 31 de diciembre de 2019 teniendo un total de 5737 observaciones, esta plataforma se encarga de brindar información de activos como, oro, plata, acciones de empresas que cotizan en la bolsa de Nueva York, divisas, etc.

Para aplicar las técnicas de predicción se divide la serie temporal en los conjuntos de entrenamiento y validación. Se usa como datos de entrenamiento del 2 de enero de 1998 al 29 de octubre de 2019 siendo 5715 datos y se deja desde el 2 de diciembre de 2019 al 31 de diciembre del mismo año para el conjunto de validación que corresponde a 22 datos.

Se realiza un análisis exploratorio de la serie y se aplica los modelos ARIMA y redes neuronales recurrentes de tipo Elman y LSTM.

Luego de realizar predicciones de los 22 días futuros utilizando las tres técnicas de predicción se calcula las medidas de error, el coeficiente U de Theil y se realiza la prueba de Diebold-Mariano para identificar la técnica que presenta predicciones más precisas.

Y por último luego de haber identificado que técnica presenta predicciones más precisas se obtiene el error porcentual absoluto medio de las 22 predicciones de dicha técnica, para identificar el límite de tiempo en donde las predicciones son eficaces.

3.1. Tipo de Investigación

La investigación es:

- Según el método de investigación es cuantitativa debido a que la variable a analizar es el índice de precio EUR/USD y los datos representa la cotización diaria del índice.
- Según el objetivo es aplicada por que la finalidad del estudio es modelar y predecir el tipo de cambio usando técnicas de predicción.
- Según el nivel de profundización en el objeto de estudio es predictiva ya que se quiere lograr una predicción al futuro usando técnicas estadísticas, en este caso se utiliza los modelos ARIMA y redes neuronales de tipo Elman y LSTM.

- Según la manipulación de variables es no experimental por que para este estudio los datos se obtuvieron de la plataforma Investing.com, esta plataforma tiene un registro diario de la cotización de diferentes activos, como el oro, plata, divisas y acciones de empresas.
- Según el tipo de inferencia es inductiva por que se pretende realizar predicciones de acuerdo a la técnica que proporcione menor error en el ajuste.
- Según el periodo temporal es longitudinal debido a que los datos están tomados en el tiempo.

3.2. Diseño de la Investigación

El presente trabajo de investigación se clasifica en un diseño no experimental puesto que los datos en estudio se obtuvieron de los registros realizados por la plataforma financiera Investing.com, es decir, que se realiza el estudio sin la manipulación de las variables y se puede observar el comportamiento de las mismas sin realizar ninguna modificación. Además, debido a que los datos están tomados en el tiempo es longitudinal, y los datos representa las cotizaciones diarias del índice del precio EUR/USD por lo tanto la investigación es cuantitativa.

3.2.1. Operacionalización de la variable

Tabla 1-3. Operacionalización de la variable en estudio

Código	Variable	Tipo/Escala
EUR/USD	Precio diario de cotización tipo de cambio EUR/USD.	Cuantitativa/Intervalo

Fuente: Investing.com (Datos históricos EUR/USD)

Realizado por: Alcoser Manya, Willian. 2020

3.2.2. Localización del Estudio

El tipo de cambio euro dólar se puede encontrar y comercializar en el mercado de divisas o también denominado FOREX, es una plataforma digital en donde se puede ingresar desde diferentes partes del mundo, el precio del euro en base al dólar se establece por diferentes factores que ocurren en los países que emiten las monedas, por ejemplo: tasa de desempleo, tasa de inflación, PIB, problemas políticos, etc.

3.2.3. Población de estudio

Para el estudio se tomaron en consideración 5737 datos que fueron registrados en la plataforma de inversiones Investing.com, desde el 02 de enero de 1998 hasta el 31 de diciembre de 2019.

3.2.4. Técnicas de recolección de datos

Los datos se obtuvieron del registro de la cotización diaria del índice de precio EUR/USD, que se encuentra ubicada en la plataforma Investing.com. La página presenta registros a partir del 02 de enero de 1998 hasta la actualidad (31 de diciembre de 2019), y diariamente se actualiza la información.

3.2.5. Recolección de información

La información fue obtenida de los registros realizados por la plataforma de inversiones Investing.com, esta página es una plataforma virtual en donde se presenta datos de las cotizaciones de activos de los mercados financieros del mundo como acciones, índices, futuros, divisas, bonos, etc. Además, es accesible a todo el público en general. Los datos vienen dados de lunes a viernes.

3.2.6. Análisis Estadístico

En el proyecto de investigación se trabajó con una serie temporal con 5737 observaciones, cada observación representa a la cotización diaria del tipo de cambio EUR/USD de lunes a viernes, los datos se obtuvieron de los registros realizados por la plataforma financiera Investing.com, se hizo un análisis exploratorio de datos graficando la serie temporal, obteniendo las medidas de tendencia central, los cuartiles y realizando un histograma. Se divide la serie en conjuntos de entrenamiento y validación y se aplica los modelos ARIMA y redes neuronales recurrentes de tipo Elman y LSTM para predecir 22 días al futuro.

Los modelos autorregresivos integrados de promedio móviles ARIMA tienen como finalidad encontrar el mejor ajuste a series de tiempo estacionarias y no estacionarias. Si la serie es no estacionaria se realiza diferencias de orden d con el objetivo de eliminar fuentes no estacionarias, la formula general de un modelo ARIMA con componentes p , d , q es:

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d Y_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q)\varepsilon_t$$

Donde ϕ_p y θ_q son los coeficientes a estimar de la parte autorregresiva y de medias móviles, $(1 - B)^d Y_t$ representa a la d -ésima diferencia de Y_t y ε_t es un ruido blanco con media 0 y varianza constante.

Para identificar uno o más modelos tentativos se utiliza la función de autocorrelación simple y parcial y se analiza el comportamiento de las mismas, luego se valida el modelo verificando los supuestos de Normalidad, Independencia, Estacionariedad y Homocedasticidad, en el caso de no cumpla un supuesto, no se recomienda realizar predicciones ya que los resultados no serían confiables.

Las redes neuronales recurrentes poseen una especie de memoria que guardan información relevante del proceso para utilizarlo en el futuro, es por eso que ha tenido un gran éxito en series temporales. Antes de ingresar los datos a la red se divide en conjuntos de entrenamiento y validación, luego se recomienda realizar un cambio de escala con la siguiente fórmula:

$$z_i = \frac{y_i - \min(Y)}{\max(Y) - \min(Y)}$$

La red neuronal recurrente de tipo Elman posee memoria a corto plazo y la de tipo LSTM posee memoria a corto y largo plazo, en ambas redes se debe determinar el número de capas, el número de neuronas, función de optimización, funciones de activación, etc, con las que se obtienen mejores resultados en sus predicciones.

Luego de predecir los 22 días a futuro de calcula los diferentes tipos de errores, el coeficiente U de Theil y se realiza la prueba de Diebold-Mariano para determinar la técnica que proporciona predicciones más precisas. Luego se calcula el error porcentual absoluto medio de las predicciones con mayor precisión y se identifica el límite de tiempo en donde las predicciones tienen menores porcentajes de error.

Para la realización de este trabajo de investigación se utilizó R-3.5.3 y Python 2.7.13 ya que son softwares libres.

CAPÍTULO IV

4. RESULTADOS Y DISCUSIÓN

Una vez visto los conceptos de las técnicas de predicción procedemos a empezar con la parte práctica del trabajo, la cual se concentra en aplicar el modelo ARIMA e implementar redes neuronales recurrentes con neuronas Elman y LSTM, y valorar los resultados obtenidos de la predicción en los siguientes 22 días.

Antes de empezar a aplicar las técnicas estadísticas se realiza un estudio exploratorio de los datos.

Análisis descriptivo de la serie

Se realiza es un gráfico de la serie para observar su comportamiento, por lo tanto, en el siguiente gráfico se muestra el precio de cotización del tipo de cambio EUR/USD a nivel diario desde el 2 de enero de 1998 hasta el 31 de diciembre de 2019 teniendo un total de 5737 observaciones.

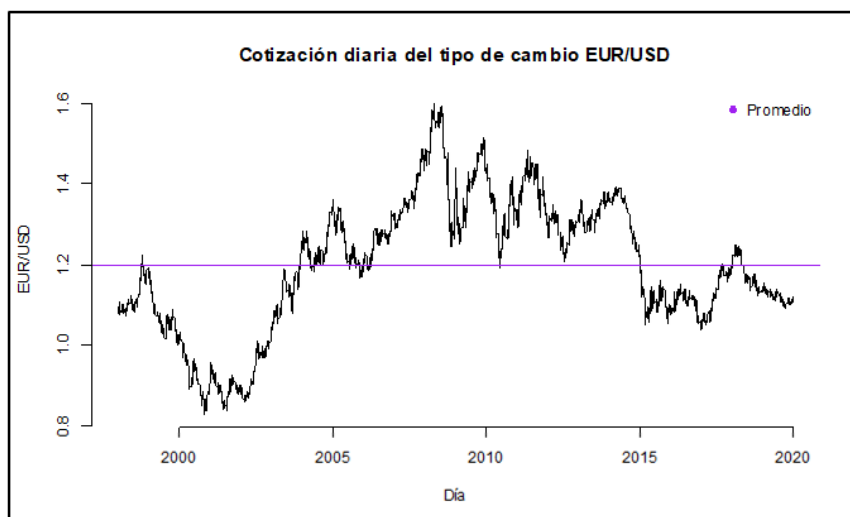


Gráfico 1-4. Serie de la cotización diaria del tipo de cambio EUR/USD

Realizado por: Alcoser, W. 2020

En el gráfico 1-4 se observa que nuestra serie temporal no sigue un patrón en el tiempo, los picos no oscilan alrededor del promedio y están alejados de ella, además, la serie no capta estacionalidad, ya que no se repite periódicamente un mismo patrón sistemático. Dada las características anteriores se podría decir que la serie temporal sigue una caminata aleatoria.

Tabla 1-4. Estadística descriptiva de la serie temporal EUR/USD

Estadístico	Valor
N	5737
Media	1.1991
Mediana	1.2014
D. Estándar	0.1628
Varianza	0.0265
Mínimo	0.8273
Máximo	1.5988

Fuente: Estadística descriptiva

Realizado por: Alcoser Manyá, Willian. 2020

En la tabla 1-4 se observa que el número de observaciones de la serie temporal es de 5737, la cotización promedio diaria es de \$1.1991, la mediana toma un valor de \$1.2014, la desviación estándar es de 0.1628, la varianza es de \$0.0265, la cotización mínima a la que llegó el tipo de cambio es de \$0.8273 y la cotización máxima es de \$1.5988.

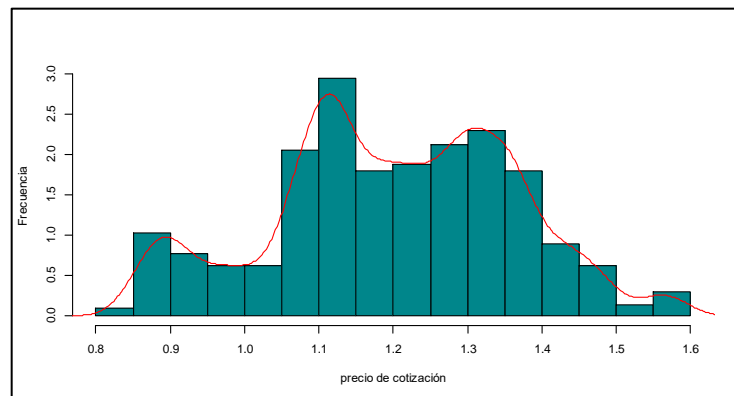


Gráfico 2-4. Histograma de la serie temporal EUR/USD

Realizado por: Alcoser, W. 2020

En el gráfico 2-4 se observa que la serie temporal se aproxima a una distribución asimétrica a la izquierda, existe una mayor concentración de datos en el intervalo de \$1.05 hasta \$1.4 aproximadamente y el ajuste de la distribución no parece ser eficiente ya que la función de densidad no se asemeja a una campana de gauss.

Una vez realizado un análisis exploratorio de la serie temporal procedemos a aplicar las técnicas de predicción, para ello todo el histórico de datos (del 2 de enero de 1998 al 31 de diciembre de 2019) se reparte en los conjuntos de entrenamiento y validación. Por otro lado, se usa como datos de

entrenamiento del 2 de enero de 1998 al 29 de octubre de 2019 (5715 datos) y dejamos desde el 2 de diciembre de 2019 al 31 de diciembre del mismo año para la validación (22 datos).

4.1. Aplicación ARIMA

Los modelos autorregresivos integrados de promedios móviles ARIMA son muy conocidos a la hora de predecir, casi parece obligatorio presentar alguna predicción con estos modelos para poder evaluar la mejora que introducen las redes neuronales recurrentes frente a estos modelos.

Se comienza este apartado analizando la estacionariedad de la serie.

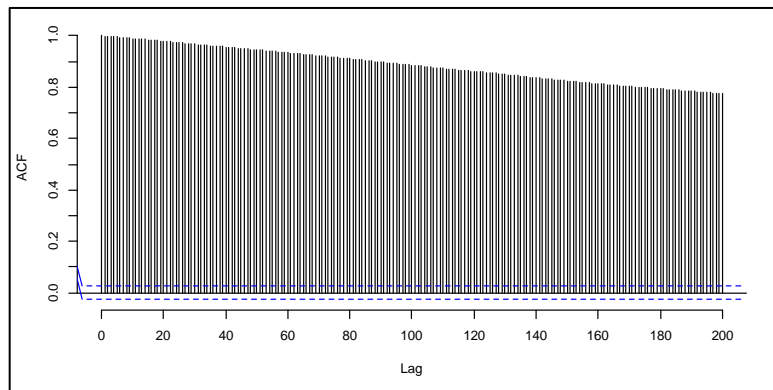


Gráfico 3-4. Función de autocorrelación simple

Realizado por: Alcoser, W. 2020

Se observa en el gráfico 3-4 que los valores de autocorrelación tienen un decrecimiento lento, este es una característica propia de las series no estacionarias. Otra forma de analizar la estacionariedad de la serie temporal es aplicando la prueba de Dickey-Fuller donde se contrasta la hipótesis nula (H_0) de que la serie no es estacionaria y la hipótesis alternativa (H_1) de que la serie es estacionaria. El nivel de significancia α establecido es de 0.05.

```
Augmented Dickey-Fuller Test
data: train
Dickey-Fuller = -1.5839, Lag order = 0, p-value = 0.7546
alternative hypothesis: stationary
```

Gráfico 4-4. Prueba de Dickey-Fuller

Realizado por: Alcoser, W. 2020

El valor p igual a 0.7546 es mayor que el nivel de significancia 0.05, por lo tanto, no existe evidencia suficiente para rechazar H_0 . Se concluye que la serie temporal no es estacionaria.

Una vez identificado que la serie no es estacionaria se procede a realizar una diferenciación de orden 1 para volverla estacionaria. Se realiza de nuevo la prueba de Dickey-Fuller a la serie diferenciada.

```
Augmented Dickey-Fuller Test
data: train_diff
Dickey-Fuller = -76.845, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary
```

Gráfico 5-4. Prueba de Dickey-Fuller de la serie diferenciada

Realizado por: Alcoser, W. 2020

El valor p 0.01 es menor que el nivel de significancia 0.05, por lo tanto, se rechaza H_0 . Se concluye que la serie diferenciada de orden 1 es estacionaria.

Se procede a realizar el gráfico de la serie diferenciada de orden 1 con su función de autocorrelación simple.

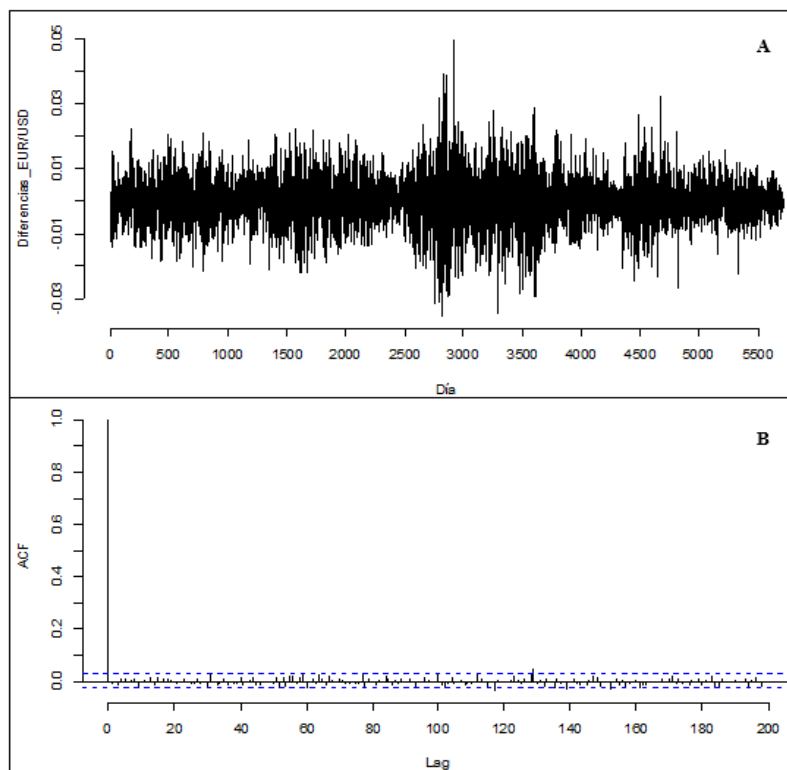


Gráfico 6-4. Serie Integrada de orden 1 y ACF

Realizado por: Alcoser, W. 2020

En el gráfico 6-4 A se observa que la serie diferenciada de orden 1 presenta las características de un ruido blanco, donde los valores son independientes e idénticamente distribuidos con media y covarianza cero, y varianza constante. En el gráfico 6-4 B (Función de autocorrelación simple) se

observa que ningún coeficiente de correlación es significativo, se puede afirmar que los datos son independientes.

Con los resultados obtenidos hasta el momento se determina un modelo de caminata aleatoria que es un modelo ARIMA(0, 1, 0) correspondiente a la siguiente expresión:

$$(1 - B)Y_t = \varepsilon_t$$

donde B es el operador de retardo, es decir, $BY_t = Y_{t-1}$ y $\varepsilon_t \sim RB(0, \sigma^2)$.

Además, se realizó la función `auto.arima()` en el software Rstudio donde determina el mejor modelo para la serie, dándome como resultado el mismo modelo.

Verificación de supuestos

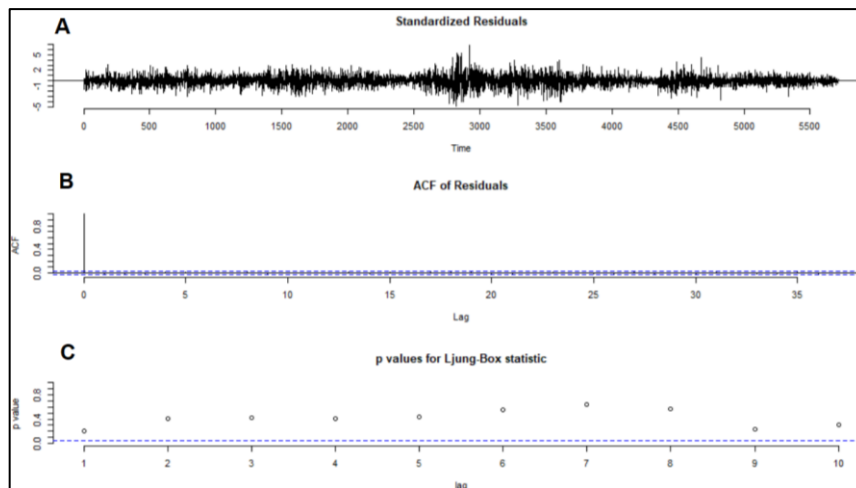


Gráfico 7-4. Diagnóstico del modelo

Realizado por: Alcoser, W. 2020

Se puede observar en el gráfico 7-4 A que los residuos del modelo cumplen con el requerimiento de estacionariedad y no presencia de tendencia, en la función de autocorrelación simple B se observa que no hay autocorrelaciones significativas y en la prueba de Ljung Box C, los valores p están por encima del valor de significancia, concluyendo que los residuos son independientes. Estos resultados se corroboran con las pruebas analíticas.

Estacionariedad

```

Augmented Dickey-Fuller Test
data: modelo$residuals
Dickey-Fuller = -76.835, Lag order = 0, p-value =0.01
alternative hypothesis: stationary
    
```

Gráfico 8-4. Prueba de Dickey-Fuller

Realizado por: Alcoser, W. 2020

Se realiza la prueba de Dickey-Fuller contrastando la hipótesis nula H_0 de no estacionariedad y la hipótesis alternativa H_1 de estacionariedad. Se obtiene un valor p de 0.01 que es menor al nivel de significancia α 0.05, por lo tanto, se rechaza H_0 y se concluye que los residuos son estacionarios.

Independencia

```
Box-Ljung test
data: modelo$residuals
X-squared = 1.5593, df = 1, p-value = 0.2118
```

Gráfico 9-4. Prueba de Box-Ljung

Realizado por: Alcoser, W. 2020

Se realiza la prueba de Box-Ljung contrastando la hipótesis nula H_0 de independencia y la hipótesis alternativa H_1 de no independencia. El valor p 0.2118 es mayor que el nivel de significancia 0.05, por lo tanto, no existe evidencia suficiente para rechazar H_0 . Se concluye que los residuos son independientes.

Normalidad

```
Lilliefors (Kolmogorov-Smirnov)
normality test
data: modelo$residuals
D = 0.046857, p-value < 2.2e-16
```

Gráfico 10-4. Prueba de Lilliefors

Realizado por: Alcoser, W. 2020

Se realiza la prueba de Lilliefors contrastando la hipótesis nula H_0 de normalidad y la hipótesis H_1 de no normalidad, se obtiene un valor p de aproximadamente cero que es menor al nivel de significancia, por lo tanto, se rechaza H_0 y se concluye que los residuos no siguen una distribución normal.

Homocedasticidad

```
Goldfeld-Quandt test
data: modelo$residuals ~ 1
GQ = 1.037, df1 = 2857, df2 = 2856, p-value = 0.1656
```

Gráfico 11-4. Prueba de Goldfeld-Quandt

Para probar este supuesto se realiza la prueba de Goldfeld-Quandt contrastando la hipótesis H_0 de homogeneidad y la hipótesis H_1 de heterocedasticidad, se obtiene un valor p de 0.1656 que es mayor al nivel de significancia, por lo tanto, no existe evidencia suficiente para rechazar H_0 y se concluye que los residuos tienen varianza homogénea.

Predicciones

Se realiza predicciones de 22 días siguientes aplicando la función $rwf()$.

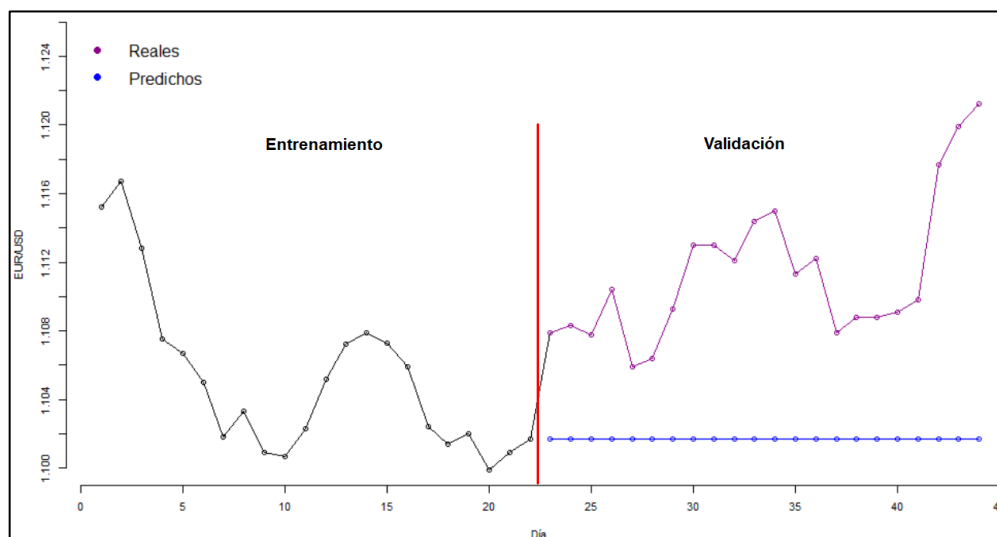


Gráfico 12-4. Valores Reales vs. Predichos – Modelo ARIMA

Realizado por: Alcoser, W. 2020

Dado que el conjunto de datos de entrenamiento es extenso se grafica una parte del 1 al 29 de noviembre de 2019 que está representado en la parte izquierda del gráfico 12-4 y en la parte derecha se encuentran los datos del conjunto de validación y las predicciones realizadas con el modelo ARIMA(0, 1, 0) de los próximos 22 días donde se identifica que las predicciones son constantes.

4.2. Aplicación Redes Neuronales Recurrentes

Preprocesamiento de los datos

El tipo de escala de los datos influyen en el procesamiento de las redes neuronales profundas especialmente cuando se usa la función de activación sigmoideal o tangente hiperbólica (Cabezón Manchado 2018).

Una estandarización alternativa es escalar las características para que se encuentren entre un valor mínimo y un valor máximo dado (en la práctica se usan los intervalos $[-1, 1]$ y $[0, 1]$) para que el valor absoluto máximo de cada variable se escale al tamaño de la unidad.

La fórmula que se utiliza para escalar una variable aleatoria y_i es:

$$z_i = \frac{y_i - \min(Y)}{\max(Y) - \min(Y)}$$

Donde $\min(Y)$ y $\max(Y)$ son valores mínimos y máximos del vector Y y z_i está escalado entre 0 y 1.

La siguiente formula se usa para transformar los datos estandarizados a la escala original:

$$y_i = z_i * (\max(Y) - \min(Y)) + \min(Y)$$

Una vez preparado los datos para ser utilizados se procede a construir las redes neuronales recurrentes. En este caso se va a implementar redes usando células de Elman y LSTM (Long Short Term Memory). Dentro de cada red hay infinitud de parámetros con los que experimentar hasta obtener mejores resultados ya que se puede jugar con las funciones de activación, con las funciones de pérdidas a minimizar, con del número de capas, con el número de neuronas en cada capa, etc. En este trabajo de investigación se detallará las redes que mejores resultados han proporcionado.

Antes de construir las redes se detallará el funcionamiento de la técnica de Dropout.

Dropout es un método que elimina o desactiva neuronas en una red neuronal con una cierta probabilidad, este parámetro toma valores de 0 a 1, si los valores son cercanos a 0 se desactivará menos neuronas y si el valor es cercano a 1 desactivará más variables.

Esta técnica ayuda a evitar el sobreentrenamiento debido a que una capa totalmente conectada ocupa la mayor cantidad de parámetros y por lo tanto las neuronas desarrollan codependencia entre ellas, lo que restringe la potencia individual de cada una y produce el sobreentrenamiento (Cabezón Manchado 2018).

Para este estudio el Dropout toma un valor de 0.2 debido a que Cabezón Manchado (2018) en su estudio concluye que es el que mejor funciona. Además, se probó las redes neuronales sin utilizar la técnica de Dropout y se obtuvieron peores resultados.

Cuando se utiliza una serie temporal para realizar pronósticos de varios pasos debe dividirse en muestras con componentes de entrada (*inputs*) y salida (*outputs*).

Tanto los componentes de inputs como de outputs estarán compuestos por múltiples pasos de tiempo y pueden tener o no el mismo número de pasos. Por ejemplo, dada la serie temporal univariante:

$$Y = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$

Podríamos usar los últimos tres pasos de tiempo como entrada y pronosticar los dos próximos pasos de tiempo, las muestras quedarían de la siguiente forma:

		<i>inputs</i>	<i>outputs</i>
muestra 1	→	[1, 2, 3]	[4, 5]
muestra 2	→	[2, 3, 4]	[5, 6]
muestra 3	→	[3, 4, 5]	[6, 7]
muestra 4	→	[4, 5, 6]	[7, 8]
muestra 5	→	[5, 6, 7]	[8, 9]
muestra 6	→	[6, 7, 8]	[9, 10]

4.2.1. Célula Elman

El primer modelo recurrente que se realiza es una red constituida con células de Elman.

```
fit_elman <- elman(x = inputs,
  y = outputs,
  size = c(3,2),
  maxit = 7000,
  learnFuncParams = c(0.2))
```

Gráfico 13-4. Red con células de Elman

Realizado por: Alcoser, W. 2020

Esta red consta de 110 pasos de tiempo como *inputs* para pronosticar los próximos 22 pasos de tiempo (*outputs*), posee dos capas con 3 y 2 neuronas cada una, un maxit (número de iteraciones para entrenar el modelo) de 7000 y un learnFuncParams (velocidad de entrenamiento de la red) de 0.2.

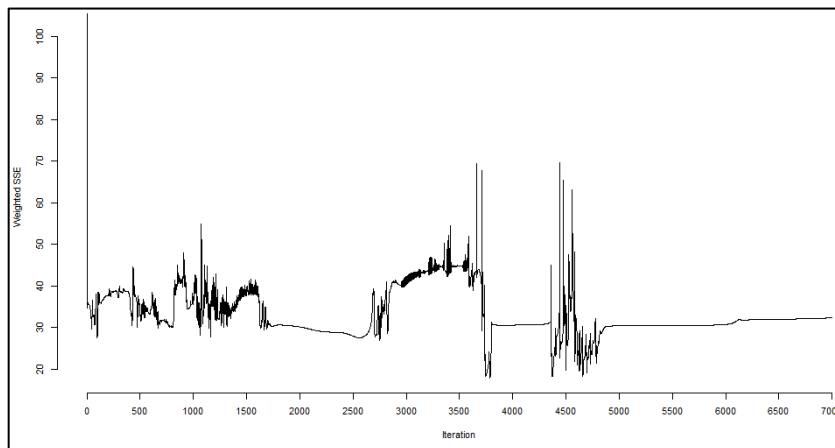


Gráfico 14-4. Comportamiento del error de la red de Elman

Realizado por: Alcoser, W. 2020

En el gráfico 14-4 se representa el comportamiento del error de la red con células Elman, se observa que desde la primera iteración hasta la iteración 5000 el error tiene un comportamiento aleatorio y a partir de la iteración 5000 el error se estabiliza o se mantiene.

Predicciones

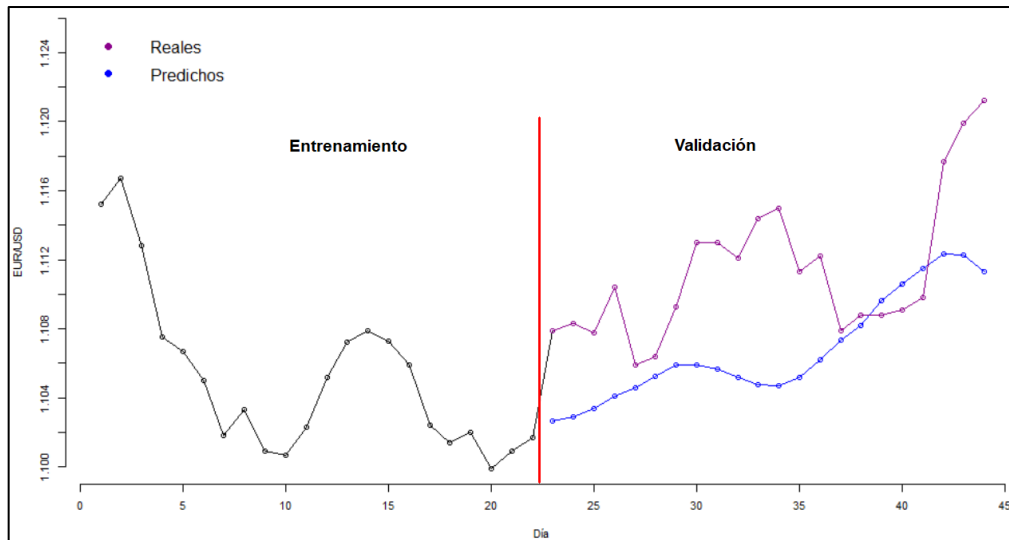


Gráfico 15-4. Valores reales vs. Predichos – RNR de Elman

Realizado por: Alcoser, W. 2020

En la parte izquierda del gráfico 15-4 se encuentran los datos del conjunto de entrenamiento únicamente del 1 al 29 de noviembre de 2019, en la parte derecha se encuentran los datos del conjunto de validación y las predicciones de los próximos 22 días utilizando la red de Elman, donde se identifica que las predicciones tienen un comportamiento ascendente del primer día de predicción al último día.

4.2.2. Célula LSTM

La segunda red construida en la investigación está constituida con células LSTM.

```
model.add(LSTM(50,
               activation = "relu",
               return_sequences = True,
               input_shape = (n_steps_in, n_features),
               use_bias = True))
model.add(LSTM(50, activation = "relu"))
model.add(Dropout(0.2))
model.add(Dense(n_steps_out))
model.compile(optimizer = 'adam', loss = 'mae')
```

Gráfico 16-4. Red con células LSTM

Realizado por: Alcoser, W. 2020

Esta red posee dos capas ocultas con 50 unidades cada una y una función de activación *relu*, y una capa de salida (Dense), en la primera capa las neuronas tienen salidas tridimensionales (*return_sequences = True*), variables de entrada (*input_shape*) y se usa el termino independiente sesgado (*use_bias = True*). En este caso se usa como función de pérdida para la red el error absoluto medio (*mae*) y como optimizador *adam*.

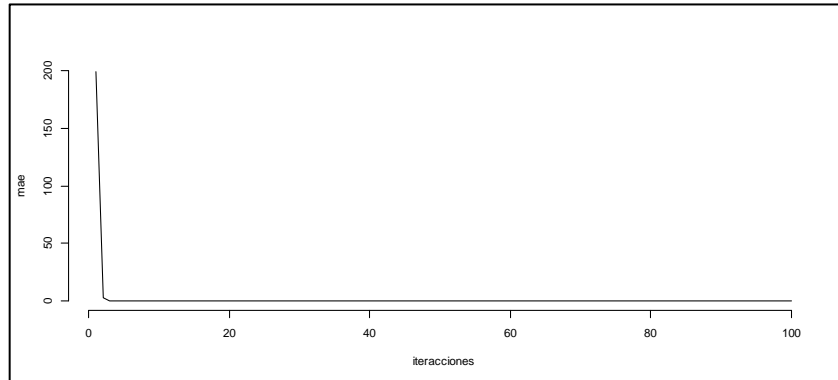


Gráfico 17-4. Comportamiento del error de la red LSTM

Realizado por: Alcoser, W. 2020

En el gráfico 17-4 se observa el comportamiento del error en función al número de iteraciones, aproximadamente el error de la iteración uno y dos es grande y a partir de la iteración tres el error se estabiliza o se mantiene.

Predicciones

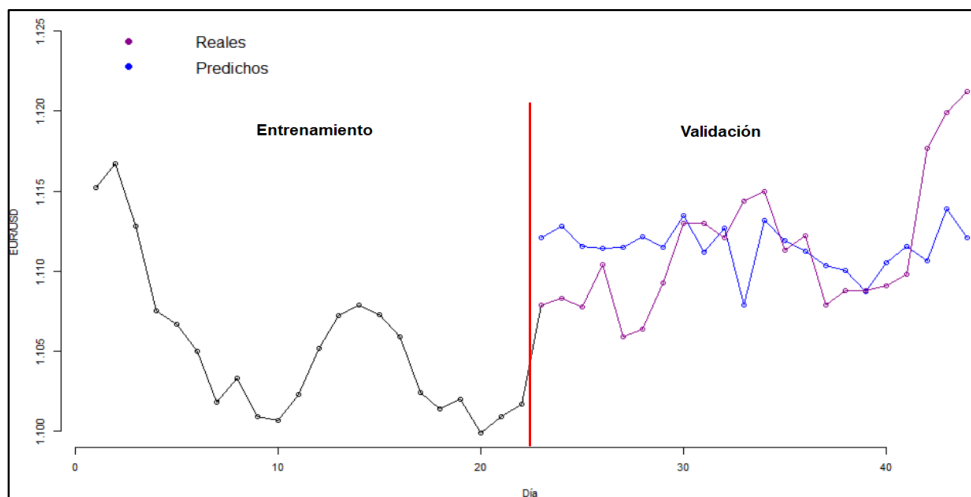


Gráfico 18-4. Valores reales vs. Predichos – RNR LSTM

Realizado por: Alcoser, W. 2020

Al igual que en los gráficos 12-4 y 15-4 en la parte izquierda se encuentran una parte de los datos del conjunto de entrenamiento del 1 al 29 de noviembre de 2019, en la parte derecha se encuentran los datos del conjunto de validación y las predicciones de los próximos 22 días utilizando la red LSTM donde se identifica que las predicciones presentan aleatoriedad.

4.3. Comparación de las técnicas de predicción

4.3.1. Caracterización de las técnicas

Los métodos estadísticos clásicos y Redes Neuronales Artificiales son técnicas de regresión que se han venido utilizando para analizar el comportamiento de una serie de tiempo, siendo el segundo el más utilizado en la última década por su amplio potencial en el pronóstico de las series, debido a sus capacidades de adaptabilidad, generalización, aprendizaje y relaciones no lineales. Ambos métodos presentan terminologías y características diferentes que se describen a continuación. Cabe mencionar que este estudio no se realizó con el fin de desprestigiar una técnica sino para compararlas y quien sabe en el futuro (si ya es que existe) poder fusionarlas.

- Las aplicaciones de las redes neuronales son consideradas como técnicas no paramétricas cuyo modelo se entrena a través de algoritmos no lineales, y los modelos ARIMA como técnicas paramétricas y el modelo se entrena a través de regresiones lineales.
- Las RNA son modelos auto adaptables que no requieren de supuestos a priori de la serie y por su flexibilidad para construir topologías de los modelos es fácil identificar y pronosticar los patrones de comportamiento en las series, en el caso de los modelos ARIMA se exhiben patrones de comportamientos como tendencia y ciclos o estacionalidades y aleatoriedad (Menacho 2014).
- EL modelo ARIMA es sensible a datos atípicos convergiendo a un modelo de ruido blanco. En el caso de las RNA como el entrenamiento es punto a punto, si la cantidad de datos atípicos es menor, entonces los datos correctos arreglarán el error generado por los datos atípicos convergiendo al modelo exacto (Ríos 2008).
- En los modelos ARIMA se denominan parámetros a los valores a estimar y en las RNA pesos.

A continuación se realiza una comparación gráfica y analítica de las predicciones realizadas con el modelo ARIMA(0, 1, 0) y redes neuronales recurrentes de tipo Elman y LSTM.

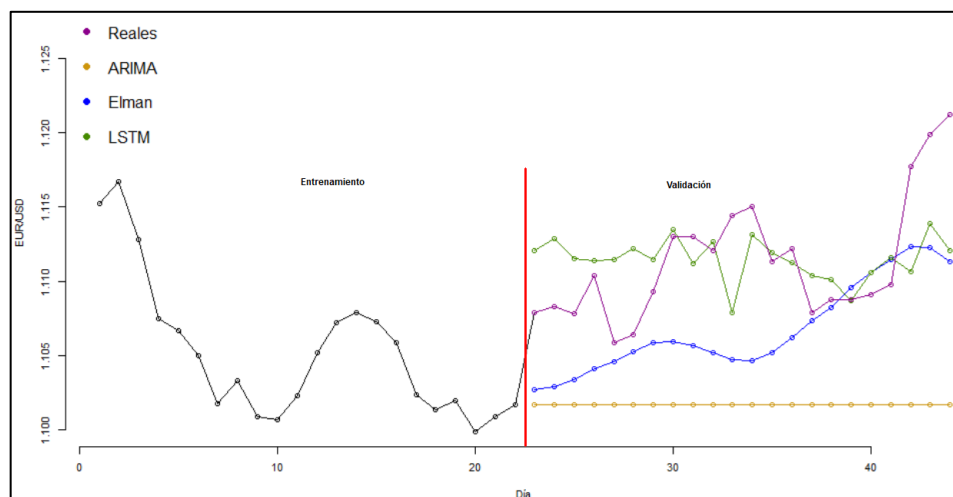


Gráfico 19-4. Predicciones realizadas con: ARIMA, Elman y LSTM

En el gráfico 19-4 se observa los datos del conjunto de validación y las predicciones de 22 días futuros utilizando el modelo ARIMA(0, 1, 0) y redes neuronales recurrentes de Elman y LSTM.

Las predicciones obtenidas con el modelo ARIMA permanecen constantes desde el primer día hasta el último día, en la red de Elman presenta tendencia alcista en sus predicciones y en la red LSTM las predicciones presentan un comportamiento aleatorio. Se realiza la evaluación de predicciones calculando las medidas de error de predicción, el coeficiente U de Theil y se realiza la prueba de Diebold-Mariano.

4.3.2. Medidas de evaluación de predicciones

Tabla 2-4. Medidas de evaluación de predicciones

Modelo	Errores dependientes de la escala				Porcentajes de error					
	MSE	RMSE	MAE	MdAE	MAPE	sMAPE	RMSPE	RMdSPE	MdAPE	sMdAPE
ARIMA(0,1,0)	0.000110	0.010496	0.00967	0.0084	0.869%	0.218%	0.941%	0.757%	0.756%	0.189%
ELMAN	3.39e-05	0.00582	0.0049	0.0053	0.44%	0.111%	0.522%	0.483%	0.483%	0.12%
LSTM	1.61e-05	0.004	0.0031	0.002	0.28%	0.071%	0.360%	0.181%	0.181%	0.045 %

Fuente: Tipos de errores

Realizado por: Alcoser Manya, Willian. 2020

En la tabla 2-4 se observa que los diferentes tipos de errores de predicción obtenidos mediante la red LSTM son menores a los errores obtenidos con la red de Elman y el modelo ARIMA(0, 1, 0), con estos resultados se podría decir que las células LSTM arrojan mejores pronósticos.

4.3.3. Análisis de precisión de los modelos

Para identificar la precisión de los modelos se obtiene el coeficiente U de Theil del conjunto de las predicciones obtenidas con las tres técnicas de predicción. Cuando el coeficiente es cercano a 0 se dice que no hay ningún tipo de desigualdad entre los valores observados y sus predicciones y los coeficientes cercanos a 1 se dice que hay mucha desigualdad.

Tabla 3-4. Coeficiente U de Theil – predicciones con: ARIMA, Elman y LSTM

ARIMA	ELMAN	LSTM
0.004743	0.002625	0.001808

Fuente: Coeficiente U de Theil

Realizado por: Alcoser Manya, Willian. 2020

En la tabla 3-4 se observa que los coeficientes obtenidos con las tres técnicas de predicción están cerca de cero, por lo tanto, las tres técnicas presentan predicciones confiables ya que no existe desigualdad significativa entre los valores reales y las predicciones. El coeficiente obtenido con la red LSTM es menor con un valor de 0.001808 en comparación con el coeficiente obtenido con la red Elman que presenta un coeficiente de 0.002625 y ARIMA con un coeficiente de 0.047.

Se realiza la prueba de Diebold – Mariano para comparar la precisión de predicción de las tres técnicas. La tabla 4-4 presenta los valores p de los contrastes de hipótesis a dos colas y unilateral.

two.sided (Dos colas)

H_0 : Las dos técnicas tienen la misma precisión de predicción

H_1 : Las dos técnicas no tienen la misma precisión de predicción

greater (Unilateral)

H_0 : Las dos técnicas tienen la misma precisión de predicción

H_1 : La técnica 2 es más preciso que la técnica 1

Tabla 4-4. Prueba de Diebold-Mariano “two.sided” y “greater” de las predicciones con: ARIMA, Elman y LSTM

	Técnica 1- Técnica 2	Técnica 1- Técnica 2	Técnica 1- Técnica 2
	ARIMA - Elman	ARIMA - LSTM	Elman - LSTM
two.sided	0.0001694	4.287e-05	0.0154
greater	8.468e-05	2.144e-05	0.0077

Fuente: Coeficiente U de Theil

Realizado por: Alcoser Manyá, Willian. 2020

De acuerdo a los valores p presentados en la tabla 4-4 se obtienen los siguientes resultados:

- ARIMA y ELMAN: En el caso del contraste de hipótesis a dos colas “two.sided”, el valor p 0.0001694 es menor que el nivel de significancia 0.05, se rechaza H_0 , y se concluye que las dos técnicas no tienen la misma precisión de predicción. En el caso de la hipótesis unilateral “greater” el valor p 8.468e-05 es menor que el nivel de significancia 0.05, se rechaza H_0 y se concluye que las predicciones realizadas con la técnica 2 (Elman) son más precisos que las predicciones realizadas con la técnica 1 (ARIMA).
- ARIMA y LSTM: En el caso del contraste de hipótesis a dos colas “two.sided”, el valor p 4.287e-05 es menor que el nivel de significancia 0.05, se rechaza H_0 , y se concluye que las

dos técnicas no tienen la misma precisión de predicción. En el caso de la hipótesis unilateral “greater” el valor p 2.144e-05 es menor que el nivel de significancia 0.05, se rechaza H_0 y se concluye que las predicciones realizadas con la técnica 2 (LSTM) son más precisos que las predicciones realizadas con la técnica 1 (ARIMA).

- Elman y LSTM: En el caso del contraste de hipótesis a dos colas “two.sided”, el valor p 0.0154 es menor que el nivel de significancia 0.05, se rechaza H_0 y se concluye que las dos técnicas no tienen la misma precisión de predicción. En el caso de la hipótesis unilateral “greater” el valor p 0.0077 es menor que el nivel de significancia 0.05, se rechaza H_0 y se concluye que las predicciones realizadas con la técnica 2 (LSTM) son más precisos que las predicciones realizadas con la técnica 1 (Elman).

Con los resultados obtenidos se concluye que las predicciones realizadas con la red LSTM son más precisos que las predicciones realizadas con el modelo ARIMA(0, 1, 0) y la red de Elman.

A continuación, se presenta el error porcentual absoluto medio (MAPE) de las 22 predicciones obtenidas con la red LSTM con el objetivo de identificar el límite de tiempo donde la técnica presenta menor error. Las predicciones que se realizaron fueron del 2 de diciembre al 31 de diciembre de 2019.

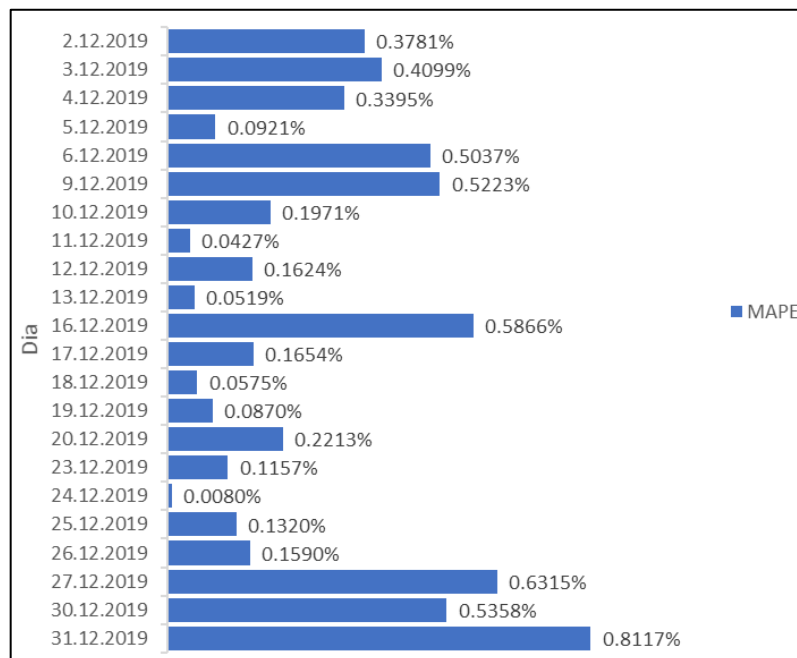


Gráfico 20-4. MAPE de las 22 predicciones

Realizado por: Alcoser, W. 2020

En el gráfico 20-4 se observa que el error porcentual absoluto medio presenta aleatoriedad, el último día de predicción presenta un porcentaje de error significativamente mayor a los demás días. En los días 2, 3, 4, 6, 9, 16, 27 y 30 de diciembre los porcentajes de error son altos, los días 5, 10, 12, 17,

20, 25 y 26 de diciembre tienen porcentajes de error bajos y en los días 11, 13, 18 y 24 de diciembre presentan porcentajes de error significativamente bajos. Estos resultados se pueden deber a la volatilidad de la serie temporal.

CONCLUSIONES

La cotización diaria del índice de precio EUR/USD tuvo un mínimo de \$0.8273 el 25 de octubre del 2000, alcanzó un máximo de \$1.60 el 22 de abril del 2008 y en promedio la cotización diaria es de \$1.199.

La serie temporal EUR/USD utilizada en el trabajo de investigación se identificó que no presenta un componente en particular como tendencia, estacionalidad o estacionariedad por lo que al realizar el respectivo análisis se determinó como camino aleatorio o *random walk*. Al realizar la primera diferencia la serie se convirtió en estacionario y se identificó un ruido blanco, es decir, media y covarianza cero y varianza constante. Se identificó un modelo ARIMA(0, 1, 0) realizando el gráfico de la serie diferenciada de orden uno con su respectiva función de autocorrelación simple y aplicando la función *auto.arima()* en Rstudio. El pronóstico utilizando este modelo en el tiempo t se determina por el valor del tiempo anterior por lo que al realizar predicciones en n pasos al futuro se genera números constantes.

La ventaja de utilizar redes neuronales artificiales es que no se necesita de supuestos a priori para determinar el comportamiento de la serie, en este caso se aplicó redes neuronales recurrentes de tipo *Elman* y *LSTM* por su eficacia al trabajar con series temporales. Ambas técnicas tienen características similares en su red (véase el ANEXO C y D) la única diferencia radica en su capacidad de memoria.

Se obtuvo predicciones de 22 días a futuro (del 2 al 31 de diciembre de 2019) utilizando las tres técnicas de predicción, se comparó gráficamente y analíticamente calculando las medidas de error, el coeficiente U de Theil y realizando la prueba de Diebold-Mariano. Con las medidas de error se identificó que las predicciones realizadas con la red LSTM presentaban errores menores a los presentados con la red de Elman y el modelo ARIMA(0, 1, 0). Con los coeficientes U de Theil obtenidos en las tres técnicas se identificó que las predicciones son confiables en los tres casos ya que sus coeficientes se acercan a cero, el menor coeficiente obtenido es de la red LSTM con un valor de 0.001808, seguido de la red de Elman con 0.002625 y por último el modelo ARIMA(0, 1, 0) con 0.004743. Y con la prueba de Diebold-Mariano se concluyó que las predicciones realizadas con la red de Elman son más precisas que las predicciones realizadas con el modelo ARIMA(0, 1, 0) y menos precisas que la red LSTM. Además, las predicciones realizadas con la red LSTM son más precisas que las realizadas con la red de Elman y el modelo ARIMA(0, 1, 0).

Una vez identificado la técnica que realiza predicciones más precisas se calcula el error porcentual absoluto medio o MAPE de las 22 predicciones (del 2 de diciembre al 31 de diciembre de 2019) observando un comportamiento aleatorio en los errores, el porcentaje de error significativamente

mayor fue obtenido el último día de predicción es decir del 31 de diciembre de 2019 y los porcentajes de error significativamente bajos correspondían a los días 11, 13, 18 y 24 de diciembre.

RECOMENDACIONES

El tiempo de modelación utilizando redes neuronales artificiales recurrentes va a depender de las características de cada red como, por ejemplo: el número de capas, número de neuronas en cada capa, etc. Por lo que no se recomienda utilizar una computadora con un procesador core i3 y memoria RAM de 4GB ya que en este trabajo se demoró entre 6 y 7 horas utilizando las características descritas en el ANEXO C y D, tardaría menos tiempo si se utiliza una computadora con procesador core i5 y una memoria RAM de 8GB.

En el desarrollo de redes neuronales artificiales se recomienda ir probando los parámetros de las redes construidas, como la función de pérdida, optimizador, learning rate, dropout, número de capas, número de neuronas, etc, ya que suelen cambiar de acuerdo al tipo de variables y a su estructura.

Debido a los resultados satisfactorios obtenidos en diferentes estudios de aplicaciones con Redes Neuronales Artificiales y Machine Learning, se recomienda enseñar este tipo de técnicas dentro de la carrera de Ingeniería Estadística Informática para que los futuros profesionales puedan asumir con eficiencia nuevos retos en esta área.

BIBLIOGRAFÍA

ARANGO, F.O. et al. “Pronóstico de los índices accionarios DAX y S&P 500 con redes neuronales diferenciales”. *Contaduría y Administración* [en línea], 2013, (México) 58(3), pp. 203-225, [Consulta: 20 diciembre 2019]. ISSN. 0186-1042. Disponible en: [http://dx.doi.org/10.1016/S0186-1042\(13\)71227-0](http://dx.doi.org/10.1016/S0186-1042(13)71227-0)

AULAMH. *Coeficiente de Theil*. [blog]. [Consulta: 18 diciembre 2019]. Disponible en: <http://aulamh.blogspot.com/2012/07/coeficiente-de-theil.html>

BOEHMKE, B. & GREENWELL, B. *Hands-On Machine Learning with R* [en línea] [Consulta: 13 diciembre 2019]. Disponible en: <https://bradleyboehmke.github.io/HOML/index.html#conventions-used-in-this-book>

CABEZÓN MANCHADO, M. Predicción demanda eléctrica española. Implementación de redes neuronales recurrentes en Python [en línea] (Tesis) (Maestría) Universidad Complutense de Madrid, España. 2018. [Consulta: 10 diciembre 2019]. Disponible en: <https://eprints.ucm.es/49444/1/2018-MIGUEL%20CABEZON%20Memoria.pdf>

CALVO, D. *Red Neuronal Recurrente - RNN*. Diego Calvo [blog]. [Consulta: 12 diciembre 2019]. Disponible en: <http://www.diegocalvo.es/red-neuronal-recurrente/>

CÉSAR, P. *Técnicas avanzadas de predicción*. Ibergarcet. Madrid, 2016, pp. 1-50

FERNÁNDEZ, S. *Series Temporales: Modelo Arima*. Universidad Autónoma de Madrid [en línea], pp. 53. Disponible en: <http://www.estadistica.net/ECONOMETRIA/SERIES-TEMPORALES/modelo-arima.pdf>

FLÓREZ, R. & FERNÁNDEZ, J. *Las Redes Neuronales Artificiales Fundamentos teóricos y aplicaciones prácticas* [en línea]. S.l.: s.n. ISBN 9788497452465. Disponible en: <https://books.google.com.ec/books?id=X0uLwi1Ap4QC&pg=PA56&lpg=PA56&dq=red+maquina+de+cauchy&source=bl&ots=gLMzelpr3j&sig=7bOm9fYQj6gwP6zZs&hl=es#v=onepage&q=red+maquina+de+cauchy&f=true>

FXCM. *¿Qué es forex?* Fxcm a leucadia company. [blog] [Consulta: 29 de noviembre 2019]. Disponible en: <https://www.fxcm.com/es/forex/que-es-forex/>

FXGM. *Historia del Forex*. www.fxgm.com. [blog] [Consulta: 03 de diciembre 2019]. Disponible en: <https://es.fxgm.com/trading-academy/forex-history/>

GÉRON, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow* [en línea] [Consulta: 4 enero 2020] Disponible en: [http://index-of.es/Varios-2/Hands on Machine Learning with Scikit](http://index-of.es/Varios-2/Hands+on+Machine+Learning+with+Scikit)

Learn and Tensorflow.pdf

GIRALDO GOMEZ, N. *Series de Tiempo* [en línea]. 2012. [Consulta: 12 diciembre 2019] Disponible en: <https://www.medellin.unal.edu.co/~ndgiraldo/Archivos Lectura/Archivos curso Series EIO/Capitulo 7 Notas de Clase.pdf>

GONZÁLEZ, P. *Análisis de series temporales económicas: modelos ARIMA* [en línea]. 2007. [Consulta: 23 diciembre 2019]. Disponible en: <https://addi.ehu.es/bitstream/handle/10810/12492/04-09gon.pdf;jsessionid=DBED3DDDAD5E8807059F6551F1A9D16D?sequence=1%0Ahttps://books.google.com.co/books?id=ftqrKnA7dTMC>

GRAPH. *Machine Learning: Qué es, tipos, ejemplos y cómo implementarlo.* [blog]. [Consulta: 19 enero 2020]. Disponible en: <https://www.grapheverywhere.com/machine-learning-que-es-tipos-ejemplos-y-como-implementarlo/>

GUJARATI, D. *Econometría*. Quinta edición. México: McGrawHill Educación, 2009, pp- 1-949

HYNDMAN, R. & ATHANASOPOULOS, G. *FORECASTING Principles and Practice* [en línea]. 2a edición. Melbourne, Australia, 2018. [Consulta: 5 enero 2020]. Disponible en: <https://otexts.com/fpp2/index.html>

IBÁÑEZ, A. & GRANADOS, R. *Semi-Supervised Learning...el gran desconocido.* [blog]. [Consulta: 19 de enero 2020]. Disponible en: <https://empresas.blogthinkbig.com/semi-supervised-learning-el-gran-desconocido/>

INVESTING.COM. *EUR/USD-Euro Dólar estadounidense.* Investing.com [blog]. Disponible en: <https://es.investing.com/currencies/eur-usd-historical-data>

KHEPRI, WILLIAM. *Redes Neuronales ¿qué son?.* [blog]. [Consulta: 4 febrero 2020]. Disponible en: <https://medium.com/@williamkhepri/redes-neuronales-que-son-a64d022298e0>

KRYGMAN, P. & OBSTFIELD, M. *Economía Internacional Teoría y Política.* 2006, pp. 295-359

KUHN, M. & JOHNSON, K. *Applied Predictive Modeling.* 2013, pp. 1-595

LEWTS, N.D. *Deep time series forecasting with Python,* 2016.

MATICH, D.J. *Redes Neuronales: Conceptos Básicos y Aplicaciones.* Historia [en línea], 2001, pp. 55. [Consulta: 4 diciembre 2019]. Disponible en: <ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf>

MAURICIO JOSÉ, A. “Análisis de series temporales”. *Revista Ingeniería Industrial* [en línea], 2007, (España) 8(1), pp. 21-33. [Consulta: 20 enero 2020]. ISSN. 0718-8307. Disponible en: <https://www.ucm.es/data/cont/docs/518-2013-11-11-JAM-IAST-Libro.pdf>

MENACHO, C. “Comparación de los métodos de series de tiempo y redes neuronales”. *Anales Científicos* [en línea], 2014, (Perú) 75(2), pp. 245-252. [Consulta: 10 enero 2020]. ISSN. 2519-7398 Disponible en: http://revistas.lamolina.edu.pe/index.php/acu/article/view/960/pdf_168

MURPHY, J. *Análisis Técnico de los mercados financieros*. Centro Libros. España, Barcelona, 2000, pp. 1-30

ORTIZ ARANGO, F. & CRUZ ARANDA, F. *Redes neuronales diferenciales*. 2011 , pp. 49-64.

PEÑA, D. *Análisis de Datos Multivariantes*, 2002, pp. 515

PILCO, V. & ACURIO, W. Técnicas estadísticas para la modelación y predicción de la temperatura y velocidad de viento en la provincia de Chimborazo [en línea] (Tesis). (Postgrado). Escuela Superior Politécnica de Chimborazo. [Consulta: 15 enero 2020]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/10955>

POZNYAK, A. et al. *Differential Neural Networks for Robust Nonlinear Control. Differential Neural Networks for Robust Nonlinear Control*. 2001, pp 1-70

REYES LORA, F. Predicción de la demanda eléctrica: comparativa arima-redes neuronales mediante software SPSS. [en línea] (Tesis) (Maestría), pp. 123. Escuela Superior de Ingenieros. [Consulta: 10 enero 2020]. Disponible en: <https://idus.us.es/xmlui/bitstream/handle/11441/48443/Fco.ManuelReyesLora-ProyectoFindeMáster.pdf?sequence=1>

RÍOS, G., 2008. Series de Tiempo. Universidad de Chile. Facultad de Ciencias Físicas y Matemáticas [en línea], pp. 52. [Consulta: 5 diciembre 2019]. Disponible en: https://www.u-cursos.cl/ingenieria/2010/1/CC52A/1/material_docente/objeto/296003

ROMERO SALDAÑA, M. “Pruebas de bondad de ajuste a una distribución normal”. *Revista Enfermería del Trabajo* [en línea], 2016, 6(3), pp. 105-114. [Consulta: 15 enero 2020]. ISSN. 2174-2510. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=5633043>

STEPHANIE. *Prueba de Goldfeld Quandt: definición, pasos para correr*. [blog]. Estadísticas cómo, 2016. [Consulta: 2 febrero 2020]. Disponible en: <https://www.statisticshowto.datasciencecentral.com/goldfeld-quandt-test/>

SWAAB, T. & TRAXLER, M. “Effects of prediction and contextual support on lexical processing: Prediction takes precedence”. *Cognition* [en línea], 2015, vol. 136, pp. 135-149. [Consulta: 4 enero 2020]. ISSN 18737838. Disponible en: <https://pubmed.ncbi.nlm.nih.gov/25497522/>

TERÁN, M. et al. Proyecto de investigación sobre la aplicación y el nivel de aceptación que presenta el foreign exchange currency o Forex dentro de la población económicamente activa de la

ciudad de Latacunga [en línea] (Tesis) (Postgrado) Escuela Politécnica del Ejército, Ecuador. 2011. pp. 1-109. [Consulta: 6 diciembre 2019]. Disponible en: <http://repositorio.espe.edu.ec/xmlui/handle/21000/4618>

TODAY, S. *Time series analysis : keras lstm deep learning - part 1 Start Learning Today ! Bussiness Science*, 2018, pp. 1-46

UNITED, T. *95% de los traders de Forex pierden dinero ¿es esto real o ficción?* TRADING UNITED. [blog]. [Consulta: 02 de diciembre 2019]. Disponible en: <https://www.tradingunited.es/95-traders-de-forex-pierden-dinero-es-real-ficcion/>

VÉLEZ CORREA, J. & NIETO FIGUEROA, P. Validación de medidas de evaluación para el pronóstico de la tasa de cambio en Colombia [en línea] (Tesis) (Maestría). Colegio de Estudios Superiores de Administración, Bogotá. 2016. pp. 1-107. [Consulta: 6 febrero 2020]. Disponible en: <https://repository.cesa.edu.co/bitstream/handle/10726/1577/MFC00491.pdf?sequence=1&isAllowed=y>

VILLADA, F. et al. “Redes neuronales artificiales aplicadas a la predicción del precio del oro”. *Informacion Tecnologica* [en línea], 2016, (Colombia) 27(5), pp. 143-150. [Consulta: 7 diciembre 2019]. ISSN 07180764. Disponible en: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-07642016000500016

ZAIONTZ, C. *Prueba Diebold-Mariano*. [en línea]. [Consulta: 7 febrero 2020]. Disponible en: <http://www.real-statistics.com/time-series-analysis/forecasting-accuracy/diebold-mariano-test/>

ZAPATA, L & HUGO, D. “Predicción del tipo de cambio peso-dólar utilizando Redes Neuronales Artificiales (rna)”. *Pensamiento & Gestión* [en línea], 2008, (Colombia) (24), pp. 29-42. [Consulta: 29 noviembre 2020]. ISSN 1657-6276. Disponible en: <http://rcientificas.uninorte.edu.co/index.php/pensamiento/article/view/3476/4920>

ANEXOS

ANEXO A: Código en R (versión 3.5.3) del análisis exploratorio de la serie

```
# Librerías
library(openxlsx)
library(readxl)
library(DT)
# Lectura de datos de un documento .xlsx
datos <- read_excel("Datos_Completos.xlsx")
seriets <- ts(datos$Precio)
# Gráfico de la serie
par(bty='n', cex = 0.7)
plot(as.Date(datos$Fecha), type = "l", seriets, xlab = "Días",
     ylab = "EUR/USD", main = "Cotización diaria del tipo de cambio EUR/USD")
abline(h=mean(seriets), col='purple')
legend("topright", "Promedio", pch = 19,
      col = "purple", bty = "n")
# Estadísticas descriptivas
res <- data.frame(N = length(seriets), Media = mean(seriets), Mediana = median(seriets),
                 D.Estandar = sd(seriets), Varianza = var(seriets), Mínimo = min(seriets),
                 Máximo = max(seriets))
res# Histograma
par(lab=c(10, 10, 0.1))
hist(seriets, freq = F, xlab = "precio de cotización", ylab = "Frecuencia",
     main = "", xlim = c(0.8, 1.6), col = "turquoise4")
lines(density(seriets), col = "red")
```

ANEXO B: Código en R (versión 3.5.3) metodología Box-Jenkins

```
# Librerías
library(openxlsx)
library(readxl)
library(forecast)
library(fpp) # DF
library(stats)
library(nortest)
## Lectura de la base de datos
datos <- read_excel("Datos_Completos.xlsx")
seriets <- ts(datos$Precio)
par(bty = 'n', cex = 0.7)
plot(as.Date(datos$Fecha), type = "l", seriets, xlab = "Días",
     ylab = "EUR/USD", main = "Cotización diaria del tipo de cambio EUR/USD")
abline(h = mean(seriets), col = 'purple')
legend("topright", "Promedio", pch = 19, col = "purple", bty = "n")
## Datos de entrenamiento y validación
train <- seriets[1:(length(seriets)-22)]
test <- seriets[(length(train)+1):5737]
## Analisis de estacionariedad
acf(train, 200, main = "")
## Prueba de Dickey - Fuller
## Ho: La serie no es estacionaria
```

```

## H1: La serie es estacionaria
adf.test(train, alternative = "stationary", k = 0)
## Diferencia de orden 1
train_diff <- diff(train, 1)
## Prueba de Dickey - Fuller de la serie diferenciada
adf.test(train_diff, alternative = "stationary", k = 0)
## Gráfico de la serie diferenciada
plot(train_diff, type = "l", ylab = "Diferencias_EUR/USD", xlab = "Día")
## Función de autocorrelación
acf(train_diff, 200, main = "")
## Modelo
modelo <- auto.arima(train)
modelo
## Verificación de supuestos
tsdiag(modelo)
## Estacionariedad
## Prueba de Dickey - Fuller
## Ho: La serie no es estacionaria
## H1: La serie es estacionaria
adf.test(modelo$residuals, alternative = "stationary", k = 0)
## Independencia
## H0: Los residuos son independientes
## H1: Los residuos no son independientes.
Box.test(modelo$residuals, type="Ljung-Box")
## Normalidad
## H0: Los residuos siguen una distribución normal
## H1: Los residuos no siguen una distribución normal
lillie.test(modelo$residuals)
## Homocedasticidad
## H0: Los residuos son homogéneos
## H1: Los residuos no son homogéneos
gqtest(modelo$residuals ~ 1)
## Predicciones
predicciones_arima <- rwf(train, 22)
## Gráfica de las predicciones
par(bty='n', cex = 0.7)
plot(seriets[(length(seriets)-43):length(seriets)], type = "o",
     ylab = "EUR/USD", xlab = "Día", ylim = c(1.1, 1.125))
lines(23:44, as.vector(predicciones_arima$mean), col = "blue", type = "o")
lines(23:44, test, type = "o", col = "darkmagenta")
legend("topleft",c("Reales", "Predichos"),
     pch = 19, col = c("darkmagenta", "blue"),
     bty = "n", cex = 1.4)
## Exportar predicciones a un document .xlsx
write.xlsx(predicciones_arima, file = "1_Predicciones_ARIMA.xlsx")

```

ANEXO C: Código en R (versión 3.5.3) Red neuronal con células de tipo ELMAN

```

# Librerías
library(RSNNS)
library(quantmod)
library(forecast)
library(readxl)
library(openxlsx)
## Lectura de la base de datos
datos <- read_excel("Datos_Completos.xlsx")

```

```

seriets <- ts(datos$Precio)
par(bty = 'n', cex = 0.7)
plot(as.Date(datos$Fecha), type = "l", seriets, xlab = "Días",
      ylab = "EUR/USD", main = "Cotización diaria del tipo de cambio EUR/USD")
abline(h = mean(seriets), col = 'purple')
legend("topright", "Promedio", pch = 19, col = "purple", bty = "n")
## Datos de entrenamiento y validación
train <- seriets[1:(length(seriets)-22)]
test <- seriets[(length(train)+1):5737]
## Cambio de escala
serieN <- (train-min(train))/(max(train)-min(train))
## Pre-Procesamiento de datos
y <- as.zoo(serieN)
l <- data.frame(y)
n <- 110
for(i in 1:n){
  l[, i+1] <- Lag(y, k = i)
}
slogN <- l[-(1:n),]
## Definición de valores de inputs y outputs
inputs <- slogN[, 23:(n+1)]
outputs <- slogN[, 1:22]
## Construcción de una RNN de Elman
set.seed(1234)
fit_elman <- elman(x = inputs,
                  y = outputs,
                  size = c(3,2),
                  maxit = 7000,
                  learnFuncParams = c(0.2))
## Grafica del error
par(bty='n', cex = 0.7)
plotIterativeError(fit_elman, main = "Comportamiento del Error")
## Predicciones
predN7 <- predict(fit_elman, inputs[5605, ])
pred_elman_1 <- as.vector(predN7*(max(train)-min(train))+min(train))
# Grafico Valores Reales vs Predichos
par(bty='n', cex = 0.7)
plot(seriets[(length(seriets)-43):length(seriets)], type = "o",
      ylab = "EUR/USD", xlab = "Día", ylim = c(1.1, 1.125))
lines(23:44, pred_elman_1, col = "blue", type = "o")
lines(23:44, test, type = "o", col = "darkmagenta")
legend("topleft", c("Reales", "Predichos"),
      pch = 19, col = c("darkmagenta", "blue"),
      bty = "n", cex = 1.4)
# Exportar predicciones a un archivo .xlsx
write.xlsx(pred_elman_1, file = "2_Predicciones_ELMAN.xlsx")

```

ANEXO D: Código en Python (versión 2.7.13) de la red neuronal con células LSTM

```

## Librerias
import numpy as np
from numpy import ndarray
from numpy import array
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Dropout

```

```

import pandas as pd
import matplotlib.pyplot as plt
import math
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
## Función para crear los array de X y Y con sus respectivos pasos de tiempo
def split_sequence(sequence, n_steps_in, n_steps_out):
    X, y = list(), list()
    for i in range(len(sequence)):
        end_ix = i + n_steps_in
        out_end_ix = end_ix + n_steps_out
        if out_end_ix > len(sequence):
            break
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix:out_end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)
## Lectura de datos de un document .xlsx
data1 = pd.read_excel('Datos_Completos.xlsx')
serie = data1.Precio
len(serie)
serie.plot()
serie.shape
## Entrenamiento y validación
train = serie[0:len(serie)-22,]
test = serie[(len(train)):len(serie), ]
## Cambio de escala de los datos de entrenamiento
train = array(train)
train1 = train.reshape(train.shape[0], 1)
scaler = MinMaxScaler(feature_range=(0, 1))
trainN = scaler.fit_transform(train1)
# Cambiamos de dimension de train para ingresar a la función
train_f = trainN.reshape(trainN.shape[0], )
## Pasos de tiempo de inputs y outputs
n_steps_in, n_steps_out = 110, 22
x_train, y_train = split_sequence(train_f, n_steps_in, n_steps_out)
## reshape from [samples, timesteps] into [samples, timesteps, features]
n_features = 1
x_train_fi = x_train.reshape((x_train.shape[0], x_train.shape[1], n_features))
## modelo LSTM
model = Sequential()
model.add(LSTM(50,
                activation = "relu",
                return_sequences = True,
                input_shape = (n_steps_in, n_features),
                use_bias = True))
model.add(LSTM(50, activation = "relu"))
model.add(Dropout(0.2))
model.add(Dense(n_steps_out))
model.compile(optimizer = 'adam', loss = 'mae')
history = model.fit(x_train_fi, y_train, epochs=50)
## Comportamiento del error
plt.figure()
plt.ylabel('loss');plt.xlabel('epoch')
plt.semilogy(history.history['loss'])
## Muestra para la predicción a 22 pasos del tiempo
serie110 = trainN[(len(trainN)-n_steps_in):len(trainN)]
serie110 = serie110.reshape((1, serie110.shape[0], n_features))
## Predicciones
pred22 = history.model.predict(serie110, verbose = 0)

```

```

pred_orig = scaler.inverse_transform(pred22)
## Cambio de dimension para crear un data frame
pred22_data = pred22.reshape((pred22.shape[1], ))
pred_orig_d = pred_orig.reshape((pred_orig.shape[1], ))
datos = pd.DataFrame({"pred_trans":pred22_data})
## Exportar predicciones a un document .xlsx
datos.to_excel('3_9_sin_escalador_relu_prediccion_LSTM.xlsx', sheet_name='example')

```

ANEXO E: Código en R (versión 3.5.3) de comparación de las predicciones de las redes con células Elman y LSTM

```

## Librerias
library(readxl)
library(openxlsx)
## Lectura de datos de un documento .xlsx
datos <- read_excel("Datos_Completos.xlsx")
seriets <- ts(datos$Precio)
datos_test <- read_excel("datos_validacion.xlsx")
p_arma <- read_excel("1_Predicciones_ARIMA.xlsx")
p_elman <- read_excel("2_Predicciones_ELMAN.xlsx")
p_lstm <- read_excel("3_7_sin_escalador_relu_prediccion_LSTM.xlsx")
## Gráfico
par(bty='n', cex = 0.7)
plot(seriets[(length(seriets)-43):length(seriets)], type = "o",
      ylab = "EUR/USD", xlab = "Día", ylim = c(1.1, 1.128))
lines(23:44, datos_test$test, type = "o", col = "darkmagenta")
lines(23:44, p_arma$`Point Forecast`, type = "o", col = "darkgoldenrod3")
lines(23:44, p_elman$Pred, col = "blue", type = "o")
lines(23:44, p_lstm$pred_trans, type = "o", col = "chartreuse4")
legend("topleft",c("Reales", "ARIMA", "Elman", "LSTM"),
      pch = 19, col = c("darkmagenta", "darkgoldenrod3", "blue", "chartreuse4"),
      bty = "n", cex = 1.4)
## Medidas de error
## ARIMA
mse <- (1/length(datos_test$test)) * sum((datos_test$test - p_arma$`Point Forecast`)^2)
mae <- mean(abs(datos_test$test - p_arma$`Point Forecast`))
rmse <- sqrt(mean((datos_test$test - p_arma$`Point Forecast`)^2))
mdape <- median(abs(datos_test$test - p_arma$`Point Forecast`))
# Porcentajes de error
mapes <- mean(abs(100 * (datos_test$test - p_arma$`Point Forecast`)/datos_test$test))
smapes <- 100*(1/length(datos_test$test))*sum(abs(datos_test$test-
      p_arma$`PointForecast`)/(abs(datos_test$test) + abs(p_arma$`Point Forecast`))/2)
rmspes <- 100*sqrt((1/length(datos_test$test))*sum(((datos_test$test-
      p_arma$`PointForecast`)/datos_test$test)^2))
rmdsps <- 100 * sqrt(median(((datos_test$test - p_arma$`Point Forecast`)/datos_test$test)^2))
mdapes <- 100 * median(abs((datos_test$test - p_arma$`Point Forecast`)/datos_test$test))
smdapes <- 100*median(abs(datos_test$test-p_arma$`PointForecast`)/(abs(datos_test$test)+
      abs(p_arma$`Point Forecast`))/2)
## Elman
mse1 <- (1/length(datos_test$test)) * sum((datos_test$test - p_elman$Pred)^2)
mae1 <- mean(abs(datos_test$test - p_elman$Pred))
rmse1 <- sqrt(mean((datos_test$test - p_elman$Pred)^2))
mdape1 <- median(abs(datos_test$test - p_elman$Pred))
## Porcentajes de error
mape1 <- mean(abs(100 * (datos_test$test - p_elman$Pred)/datos_test$test))
smape1 <- 100*(1/length(datos_test$test))*sum(abs(datos_test$test-
      p_arma$`PointForecast`)/(abs(datos_test$test) + abs(p_arma$`Point Forecast`))/2)

```



```

rmspe1<-100*sqrt((1/length(datos_test$test)) * sum(((datos_test$test - p_elman$Pred)/datos_test$test)^2))
rmdspe1 <- 100 * sqrt(median(((datos_test$test - p_elman$Pred)/datos_test$test)^2))
mdape1 <- 100 * median(abs((datos_test$test - p_elman$Pred)/datos_test$test))
smdape1<-100*median(abs(datos_test$test - p_elman$Pred)/(abs(datos_test$test) + abs(p_elman$Pred))/2)
## LSTM
mse2 <- (1/length(datos_test$test)) * sum((datos_test$test - p_lstm$pred_trans)^2)
mae2 <- mean(abs(datos_test$test - p_lstm$pred_trans))
rmse2 <- sqrt(mean((datos_test$test - p_lstm$pred_trans)^2))
mdape2 <- median(abs(datos_test$test - p_lstm$pred_trans))
## Porcentajes de error
mape2 <- mean(abs(100 * (datos_test$test - p_lstm$pred_trans)/datos_test$test))
smape2<-100*(1/length(datos_test$test))*sum(abs(datos_test$test -
p_lstm$pred_trans)/(abs(datos_test$test) + abs(p_lstm$pred_trans))/2)
rmspe2<-100*sqrt((1/length(datos_test$test))*sum(((datos_test$test
- p_lstm$pred_trans)/datos_test$test)^2))
rmdspe2 <- 100 * sqrt(median(((datos_test$test - p_lstm$pred_trans)/datos_test$test)^2))
mdape2 <- 100 * median(abs((datos_test$test - p_lstm$pred_trans)/datos_test$test))
smdape2<-100*median(abs(datos_test$test-p_lstm$pred_trans)/(abs(datos_test$test)+
abs(p_lstm$pred_trans))/2)

```

ANEXO F: código en R (versión 3.5.3) del Coeficiente U de Theil

```

## Librerias
library(readxl)
library(openxlsx)
## Lectura de datos de un documento .xlsx
datos_test <- read_excel("datos_validacion.xlsx" )
p_arima <- read_excel("1_Predicciones_ARIMA.xlsx" )
p_elman <- read_excel("2_Predicciones_ELMAN.xlsx" )
p_lstm <- read_excel("3_7_sin_escalador_relu_prediccion_LSTM.xlsx")
## Coeficiente U de Theil de ARIMA
u_arima1 <- sqrt(mean((datos_test$test - p_arima$`Point Forecast`)^2))/(sqrt(mean(datos_test$test^2)) +
sqrt(mean(p_arima$`Point Forecast`^2)))
## Coeficiente U de Theil de la Elman
u_elman <- sqrt(mean((datos_test$test - p_elman$Pred)^2))/(sqrt(mean(datos_test$test^2)) +
sqrt(mean(p_elman$Pred^2)))
## Coeficiente U de Theil de la red de LSTM
u_lstm <- sqrt(mean((datos_test$test - p_lstm$pred_trans)^2))/(sqrt(mean(datos_test$test^2)) +
sqrt(mean(p_lstm$pred_trans^2)))

```

ANEXO G: Código en R (versión 3.5.3) de la Prueba de Diebold-Mariano

```

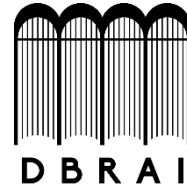
## Librerias
library(readxl)
library(openxlsx)
## Lectura de datos de un documento .xlsx
datos_test <- read_excel("datos_validacion.xlsx" )
p_arima <- read_excel("1_Predicciones_ARIMA.xlsx" )
p_elman <- read_excel("2_Predicciones_ELMAN.xlsx" )
p_lstm <- read_excel("3_7_sin_escalador_relu_prediccion_LSTM.xlsx")
## Diferencia de los datos originales con los predichos

```

```
e_ar <- datos_test$test - p_arima$`Point Forecast`
e_elman <- datos_test$test - p_elman$Pred
e_lstm <- datos_test$test - p_lstm$pred_trans
## Prueba Unilateral "two.sided"
## Ho: Los dos métodos tienen la misma precisión de pronóstico
## H1: El método 2 es más preciso que el método 1
dm.test(e_ar, e_elman, alternative = "two.sided", h = 1)
dm.test(e_ar, e_lstm, alternative = "two.sided", h = 1)
dm.test(e_elman, e_lstm, alternative = "two.sided", h = 1)
## Prueba a dos colas "greater"
## Ho: Los dos métodos tienen la misma precisión de pronóstico
## H1: El método 1 y el método 2 tienen diferentes niveles de precisión
dm.test(e_ar, e_elman, alternative = "greater", h = 1)
dm.test(e_ar, e_lstm, alternative = "greater", h = 1)
dm.test(e_elman, e_lstm, alternative = "greater", h = 1)
```



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
DIRECCIÓN DE BIBLIOTECAS Y RECURSOS PARA EL
APRENDIZAJE Y LA INVESTIGACIÓN



UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 30 / 07 / 2020

INFORMACIÓN DEL AUTOR/A (S)
Nombres – Apellidos: Willian Danilo Alcoser Manyá
INFORMACIÓN INSTITUCIONAL
Facultad: Ciencias
Carrera: Ingeniería en Estadística Informática
Título a optar: Ingeniero En Estadística Informática
f. Analista de Biblioteca responsable: Lic. Luis Caminos Vargas Mgs.



30-07-2020

0154-DBRAI-UPT-2020