

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE COMPUTACIÓN

*Trabajo de titulación previo
a la obtención del título de Ingeniero
en Ciencias de la Computación*

ARTÍCULO ACADÉMICO:

**“DISEÑO Y DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN
BASADO EN FILTRADO COLABORATIVO UTILIZANDO DATOS
SECUENCIALES MEDIANTE REDES NEURONALES RECURRENTE”**

AUTOR:

BRYAM DAVID VEGA MORENO

TUTOR:

REMIGIO ISMAEL HURTADO ORTIZ, Ph.D.

CUENCA - ECUADOR

2021

CESIÓN DE DERECHOS DE AUTOR

Yo, Bryam David Vega Moreno con documento de identificación N° 0150749059, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del trabajo de titulación: **“DISEÑO Y DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN FILTRADO COLABORATIVO UTILIZANDO DATOS SECUENCIALES MEDIANTE REDES NEURONALES RECURRENTE”**, mismo que ha sido desarrollado para optar por el título de: *Ingeniero en Ciencias de la Computación*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, octubre de 2021.



Bryam David Vega Moreno

C.I. 0150749059

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **“DISEÑO Y DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN FILTRADO COLABORATIVO UTILIZANDO DATOS SECUENCIALES MEDIANTE REDES NEURONALES RECURRENTE”**, realizado por Bryam David Vega Moreno, obteniendo el *Artículo Académico*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, octubre de 2021.



Remigio Ismael Hurtado Ortiz, Ph.D.

C.I. 0104621388

DECLARATORIA DE RESPONSABILIDAD

Yo, Bryam David Vega Moreno con documento de identificación N° 0150749059, autor del trabajo de titulación: **“DISEÑO Y DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN FILTRADO COLABORATIVO UTILIZANDO DATOS SECUENCIALES MEDIANTE REDES NEURONALES RECURRENTE”**, certifico que el total contenido del *Artículo Académico*, es de mi exclusiva responsabilidad y autoría.

Cuenca, octubre de 2021.



Bryam David Vega Moreno

C.I. 0150749059

DEDICATORIA

Dedico este trabajo a mis padres Vinicio Vega y Luz Moreno, ellos han sido el motor que me ha impulsado a seguir adelante en mi vida profesional y personal, gracias por su apoyo y cariño incondicional y siempre estar a mi lado en cada momento, sin ellos nada de esto hubiese sido posible, este trabajo es logro mío y suyo.

A mi hermano Juan Vega por su apoyo en cada momento de mi carrera, un ejemplo a seguir como hermano, amigo y profesional. A mi familia que siempre ha estado presente en cada momento importante de mi vida y siempre me ha dado animo para seguir adelante.

Por último y no menos importante, dedico a mis amigos, de manera especial a Santiago Pardo, Ismael Peñafiel, Ruben Baculima, Jason Heras, Gabriel Reinoso, Paula Cherez y Andrea Reyes por haber estado a mi lado en cada momento de mi vida profesional y personal, gracias por el apoyo incondicional que siempre he recibido de su parte desde el colegio hasta hoy, los llevo siempre en mi corazón.

Bryam David Vega Moreno

AGRADECIMIENTO

Primero quiero agradecer a Dios por mi salud y la de mis seres queridos que a lo largo de estos difíciles momentos que ha atravesado el mundo he tenido la felicidad de tenerlos a mi lado. Agradezco a la Universidad por todos los conocimientos dados a lo largo de estos 5 años en donde me han formado como un buen profesional y un buen ser humano.

Agradezco a mis padres por todo su cariño y dedicación hacia mi, su amor y responsabilidad como padres me ha permitido llegar al final de esta etapa de la mejor manera, todo este trabajo es gracias a su gran trabajo como padres y como seres humanos.

Agradezco a mis maestros que con su dedicación y esfuerzo han hecho de mi un buen profesional. De manera especial agradezco a mi tutor y maestro Remigio Hurtado, un gusto y honor haber aprendido de él como profesional y como persona. Gracias por haberme guiado a lo largo de este trabajo y darme ánimos para seguir adelante con todas mis metas.

Por último, agradezco a toda mi familia, ellos han sido pieza fundamental para mi desarrollo como profesional y personal, cada miembro de mi familia ha aportado con su grano de arena para ser la persona que soy ahora.

Bryam David Vega Moreno

RESUMEN

Los sistemas de recomendación hoy en día utilizan el filtrado colaborativo como una de las técnicas más esenciales para realizar recomendaciones. La recomendación a usuarios es un gran reto para el filtrado colaborativo ya que las compañías hoy en día tienen una gran cantidad de usuarios y en consecuencia a ello, existe una gran cantidad de esparcimiento de datos debido a que no hay muchos usuarios que voten por un producto, un programa de entretenimiento, entre otros. En adición a ello, estar al día con las preferencias y gustos del usuario convierte al filtrado colaborativo como un problema de predicción de secuencias. Tomando en cuenta dicha interpretación y a su vez los problemas existentes, se propone el diseño y desarrollo de un sistema de recomendación basado en filtrado colaborativo utilizando datos secuenciales mediante una Red Neuronal Recurrente con el objetivo de recomendar items acorde a los gustos y preferencias que el usuario presente actualmente. Con ello se busca crear un modelo inicial de red neuronal recurrente que pasara por un proceso de optimización con el objetivo de mejorar dicho modelo. Además de ello, se propone una reducción de dimensionalidad en el preprocesamiento utilizando la descomposición de valores singulares o SVD por sus siglas en ingles, con la finalidad de resolver el problema de esparcimiento que se tiene en el conjunto de datos. Con ello se evalúa el modelo propuesto con métricas de calidad tales como MAE (Mean Absolute Error) y precision score, con el objetivo de compararlo con modelos creados en este trabajo. De la misma manera, se realiza una comparación de nuestro modelo con métodos de línea base como K vecinos más cercanos, factorización matricial, entre otros, con el objetivo de demostrar la potencialidad de nuestro modelo en los sistemas de recomendación.

Palabras clave: Sistemas de recomendación, filtrado colaborativo, LSTM, deep learning, redes neuronales recurrentes, reducción de dimensionalidad.

ABSTRACT

Recommendation systems today use collaborative filtering as one of the most essential techniques for making recommendations. User recommendation is a big challenge for collaborative filtering because companies today have a large number of users and consequently, there is a large amount of data sprawl because not many users vote for a product, an entertainment program, among others. In addition, keeping up with user preferences and tastes makes collaborative filtering a sequence prediction problem. Taking into account this interpretation and the existing problems, we propose the design and development of a recommendation system based on collaborative filtering using sequential data through a Recurrent Neural Network with the objective of recommending items according to the tastes and preferences that the user currently has. With this, we seek to create an initial recurrent neural network model that will go through an optimization process in order to improve the model. In addition, a dimensionality reduction is proposed in the preprocessing using singular value decomposition or SVD, in order to solve the problem of sparsity in the data set. With this, the proposed model is evaluated with quality metrics such as MAE (Mean Absolute Error) and precision score, in order to compare it with models created in this work. In the same way, a comparison of our model with baseline methods such as K nearest neighbors, matrix factorization, among others, is performed in order to demonstrate the potential of our model in recommender systems.

Keywords: Recommender systems, collaborative filtering, LSTM, deep learning, recurrent neural networks, dimensionality reduction.

ÍNDICE DE CONTENIDO

I.	INTRODUCCIÓN	10
A.	ENFOQUES BASADOS EN MEMORIA	10
B.	ENFOQUES BASADOS EN MODELOS	11
C.	FUNDAMENTOS DE REDES NEURONALES	11
D.	FUNDAMENTOS DE SERIES TEMPORALES	12
II.	TRABAJOS RELACIONADOS.....	13
III.	METODOLOGÍA.....	16
A.	CONCEPTOS PREVIOS.....	16
B.	DISEÑO DEL MÉTODO	16
	• Fase 1: Data gathering (extracción de datos).....	18
	• Fase 2: Data wrangling	18
	• Fase 3: Diseño y optimización.....	18
	• Fase 4: Entrenamiento con el modelo de red neuronal recurrente optimizado	19
	• Fase 5: Predicciones y recomendaciones.....	19
	• Fase 6: Evaluación de predicciones y recomendaciones	20
IV.	DISEÑO DEL EXPERIMENTO.....	20
A.	EXPERIMENTOS ESTABLECIDOS	20
	a. Arquitectura de los modelos de red neuronal.....	21
	b. Proceso de optimización y fine-tuning	22
B.	MEDIDAS DE CALIDAD	24
V.	RESULTADOS Y DISCUSIÓN	25
VI.	CONCLUSIONES	28
VII.	REFERENCIAS	29

I. Introducción

Los sistemas de recomendación (SR) según (Hurtado et al., 2019) son parte “ de un conjunto actual de herramientas de inteligencia artificial para abordar problemas de sobrecarga de información” con la finalidad de resolverlos y mitigarlos. Desde la perspectiva de un usuario de SR, según (Zhu & Hurtado, 2018) “en base a sus preferencias pasadas, el sistema recomienda automáticamente una serie de elementos que estén disponibles (libros, películas, canciones, etc.) y que el usuario no haya consumido”. El SR realiza recomendaciones a partir de diferentes fuentes de información; y según la clasificación de (Zhu & Hurtado, 2018) estos pueden ser: basado en filtrado colaborativo, basados en contenidos, basados en base social, basado en información demográfica y conscientes del contexto. De dicha clasificación, los más comunes y utilizados suelen ser los basados en filtrado colaborativo y los basados en contenido.

SR **basados en contenidos** realiza la recomendación de la siguiente manera: Según (Wu et al., 2014) “si el usuario al que se le desea recomendar (usuario activo) le ha gustado un producto o un servicio, SR recomienda productos o servicios similares”. Ej.: Si a un usuario le ha gustado una canción de rock, es probable que se le recomiende discografías de bandas relacionadas a su género o géneros cercanos (metal, rock and roll, etc.) que no haya escuchado. Un fuerte inconveniente del enfoque basado en contenido es su falta de diversidad en recomendaciones.

SR **basado en filtrado colaborativo** generalmente es el más utilizado y el que mejores resultados obtiene. (Zhu & Hurtado, 2018) explica su funcionamiento: “Al usuario activo se le recomiendan artículos que no haya consumido y que hayan sido valorados positivamente por usuarios que tengan preferencias similares al usuario activo”. Usualmente, según (Zhu & Hurtado, 2018) “la información se estructura como una matriz que almacena las preferencias (explícitas o implícitas del usuario)”. Con el objetivo de extraer la información más notable de dichas matrices existen una gran variedad de enfoques, de los cuales, los más comunes son: basado en memoria y basado en modelos.

A. Enfoques basados en memoria

Uno de los métodos más comunes de este enfoque es el algoritmo KNN (K vecinos más cercanos), cuyo objetivo es encontrar los k usuarios más similares (vecinos) para cada usuario activo, después de ello, recomienda artículos o elementos que el usuario no ha consumido y

que han sido altamente valorados por sus vecinos. Este algoritmo se alimenta de las medidas de similitud de usuario a usuario. Las medidas más tradicionales son: Correlación de Pearson, Coseno y Jaccard. Este método usualmente suele ser uno de los más utilizados en los SR por sus buenas predicciones y recomendaciones.

B. Enfoques basados en modelos

Actualmente, según (J. Bobadilla et al., 2013) los SR basado en FC se suelen diseñar utilizando el enfoque basado en modelos. Dicho enfoque crea un modelo a partir de los datos, y luego de ello, se obtienen las recomendaciones del modelo. Para (Yu et al., 2017) el método más tradicional relacionado a este enfoque es la factorización matricial (FM). El funcionamiento de este método lo explica (Guan et al., 2017): “Las matrices de clasificaciones dispersas se comprimen en dos matrices de factores densos: 1) Una matriz que contiene la información de los usuarios cuyo tamaño está dado por (usuarios-factores) y 2) Otra que contiene la información de los elementos cuyo tamaño es (elementos-factores)”. El número de factores según (Zhu & Hurtado, 2018) suele ser pequeño (entre 10 a 50) obteniendo como resultado dos matrices cuyo tamaño es mucho menor que el tamaño de la información general (usuario-elementos). De acuerdo con (McNee et al., 2006) y (Said & Bellogín, 2014) “el proceso de predicción y recomendación mejora la calidad, a través de enfoques basados en memoria”.

Si bien estos métodos que utilizaran como línea base, dan buenos resultados, están lejos de ser perfectos y las mejoras se están produciendo más lentamente según (Devooght & Bersini, 2016). Además de ello, no están adaptados a la evolución de los interés y gustos de los usuarios. **Por dicha razón, nace la necesidad de buscar nuevas alternativas y metodologías, con la finalidad de adaptar nuestros SR a la evolución de gustos y preferencias del usuario. En la actualidad, se están buscando nuevas técnicas de recomendación utilizando Redes Neuronales (RN) con el fin de mejorar la precisión de los SR que las empresas actualmente desarrollan.**

C. Fundamentos de redes neuronales

Una red neuronal se puede caracterizar por el modelo de la neurona, su esquema de conexión que presentan las neuronas, entre otras palabras, su topología y algoritmo de aprendizaje que se utiliza para adaptar su función a las necesidades del problema. La topología de una red neuronal no es más que la manera en cómo se conectan las neuronas con otras y la

cantidad de neuronas que describen la red. Con el pasar del tiempo según (Cruz et al., 2007) “se ha producido una amplia variedad de topologías de redes neuronales, sin embargo la mayoría de ellas se divide en dos grandes grupos: Las redes multicapa (feed-forward) y las redes neuronales recurrentes”.

Las *redes feed-forward* no tienen ciclo. Según (Cruz et al., 2007) “generalmente estas redes son denominadas estáticas, pues producen una única salida para un conjunto de entrada, o sea, el estado de una red es independiente del estado anterior”.

Las *redes neuronales recurrentes* (RNR) son sistemas dinámicos. Para (Cruz et al., 2007) “las RNR son capaces de realizar una amplia variedad de tareas computacionales incluyendo tratamiento de secuencias, la continuación de una trayectoria, la predicción no lineal y la modelación de sistemas dinámicos”.

Tomando en consideración el problema de los métodos base referente a la evolución de gustos y preferencias del usuario, se puede tomar al FC como un problema de predicción de secuencias. Con ello, las redes neuronales recurrentes son la mejor opción para resolver dicho problema ya que estas trabajan muy bien con datos secuenciales. Un tipo de dato secuencial muy utilizado y comúnmente presente en los conjuntos de datos es el tiempo, el cual forma parte de las series temporales.

D. Fundamentos de series temporales

Para (Catalán, n.d.) una serie temporal es: “Una colección de observaciones de una variable realizadas de forma secuencial en el tiempo, en las que el orden de observación es importante”. Una característica muy importante de las series temporales es que van ligadas a instantes de tiempo. Por ello, en el ámbito de los SR, estas series temporales permiten dar continuidad al conjunto de datos, de manera que se mantenga ordenada y actualizada, y se pueda realizar recomendaciones adaptadas a las preferencias y gustos del usuario combinando las RNR y las series temporales.

Otro problema de los SR basado en FC es el esparcimiento de datos. Al tener una gran cantidad de datos y valores vacíos por usuarios que no califican ciertos elementos, se produce un sesgo en los datos, haciendo que las recomendaciones no sean exactas. Para resolver este problema se implementan técnicas de reducción de dimensionalidad (RD).

Según (Moreno et al., 2019) “la reducción de dimensionalidad permite generar nuevas relaciones a partir de características similares de un conjunto de datos.” Entre estos métodos tenemos la descomposición de valores singulares (SVD). SVD consiste en obtener un conjunto de factores que caractericen a uno o varios datos. Dicha técnica se utiliza en SR para obtener factores que caractericen a cada uno de los usuarios y elementos del conjunto de datos, haciendo que cada uno tenga una importancia dentro del mismo.

Tomando en cuenta todos los problemas antes mencionados y sus posibles soluciones, en este proyecto se plantea un nuevo SR basado en FC utilizando datos secuencias mediante RNR. Este nuevo método consiste en realizar un proceso de reducción de dimensionalidad mediante SVD con el fin de evitar sesgo en los datos, para luego de ello, utilizar el tiempo como característica fundamental para el entrenamiento y prueba de un modelo inicial de RNR el cual será optimizado y comparado con los métodos de línea base utilizando métricas de calidad tales como el MAE (*Mean Absolute Error*) y *Precision Score*, con el objetivo de demostrar la potencialidad del método propuesto.

Dicho trabajo se plantea debido a que no existen muchos proyectos relacionados a SR utilizando redes neuronales con datos secuenciales. Este sistema se realiza para proponer un nuevo método que resuelve problemas de evolución de preferencias y gustos de los usuarios hacia ciertos elementos.

II. Trabajos relacionados

El filtrado colaborativo como una de las técnicas de recomendación personalizada más comunes, se ha utilizado ampliamente en muchos dominios. En (Yang et al., 2016) se presenta un sistema de recomendación con el objetivo de recomendar los intereses de los usuarios móviles tomando en cuenta datos de los usuarios incluyendo sus valoraciones y su comportamiento. En (Zhu & Hurtado, 2018) presentan un nuevo método de FC con el objetivo de mejorar las predicciones de enfoques basados en memoria logrando mejorar los resultados de precisión que existen. Dicho método logra “la mejora de la precisión combinando la relevancia numérica de las calificaciones con información no numérica basada en la estructura de votos” explica (Zhu & Hurtado, 2018).

Por otro lado, se han implementado nuevas técnicas combinando el filtrado colaborativo con otros enfoques. Por ejemplo, en (Moreno et al., 2019) se propone un nuevo sistema de recomendación híbrido combinando los enfoques de filtrado colaborativo y filtrado basado en contenido para recomendar un grupo de elementos a un grupo específico de usuarios. El presente trabajo consiste en utilizar el filtrado basado en contenido para agrupar a los usuarios (activos) en grupos de usuarios acorde a una característica que comparten en común. Además de ello, se han realizado varias investigaciones basadas en los enfoques que utiliza el filtrado colaborativo para extraer información.

El método KNN, el cual es el más común en *enfoques basados en memoria* tiene una gran escasez de vectores de rating provocando que las medidas de similitud tradicionales de este método no sean efectivas, por lo cual, nace la necesidad de crear nuevas medidas de similitud que toman en consideración dicho problema. En (J. Bobadilla et al., 2010) se presenta una nueva medida denominada JMSD, la cual según lo explica (J. Bobadilla et al., 2010) “combina la información numérica de los votos con información independiente de dichos valores, basada en las proporciones de votos comunes y no comunes entre cada par de usuarios”.

En la medida diseñada por (J. Bobadilla et al., 2014) proporciona resultados equilibrados y de alta calidad; además de ello, dichos resultados se complementan con un tiempo de procesamiento bajo, similar al requerido para ejecutar métricas de similitud tradicionales. Por último en (Liu et al., 2014) tenemos una nueva medida de similitud de usuarios el cual pretende mejorar el rendimiento de la recomendación cuando existen pocas calificaciones para calcular la similitud de cada usuario. La métrica presentada en este artículo no solamente considera la información del contexto local de las calificaciones de usuario, sino también las preferencias globales del comportamiento del usuario activo.

En los *enfoques basados en modelos* tenemos como método más común la factorización matricial, el cual ha ido ganando popularidad debido a su buen rendimiento en los sistemas de recomendación. En (Hernando et al., 2016) se nos presenta el uso de una FM utilizando un modelo de probabilidad Bayesiana. El método lo explica (Hernando et al., 2016) de la siguiente manera: “Consiste en factorizar la matriz de clasificación en dos matrices no negativas cuyos componentes se encuentren dentro del rango $[0,1]$ con un significado probabilístico comprensible”. Por otro lado, en (Aleksandrova et al., 2017) se nos presenta una interpretación automática de las funciones de FM como usuarios, denominada usuarios representativos. Dicha interpretación toma como base

el estudio de las matrices que resultan de la factorización y en su vínculo con la matriz de clasificación original.

Tomando en cuenta el problema de los sistemas de recomendación referente a la escasez de datos, y conociendo la gran popularidad de los métodos de factorización matricial (Guan et al., 2017) presenta “un nuevo modelo de factorización matricial, llamado Enhanced SVD (ESVD), que incorpora los algoritmos clásicos de factorización matricial con finalización de calificaciones inspirados en el aprendizaje activo”. En este trabajo se combina la técnica de factorización matricial con SVD con el fin de realizar buenas recomendaciones tomando en cuenta la escasez de los datos y así resolver el problema de sesgo y desequilibrio.

De todos los métodos antes mencionados, ninguno toma en cuenta la evolución de las preferencias y gustos del usuario activo. Por lo que en (Devooght & Bersini, 2016) se presenta un sistema de recomendación basado en filtrado colaborativo utilizando redes neuronales recurrentes. Dicho método propuesto utiliza las RNR con LSTM para adaptar el filtrado colaborativo a la evolución del gusto de los usuarios. Para ello, replantean el filtrado colaborativo como un problema de predicción de secuencias. Dicho trabajo según (Devooght & Bersini, 2016) “demuestra que las LSTM son competitivas en todos los aspectos y supera ampliamente a otros métodos en términos de Cobertura (*Coverage*) de ítems y predicciones a corto plazo”. Por otro lado en (Christakou et al., 2007) se plantea un sistema de recomendación híbrido basado en filtrado colaborativo y basado en contenido usando redes neuronales con el objetivo de realizar recomendaciones precisas sobre películas. Dicho trabajo combina los resultados del filtrado mediante operadores de agregaciones booleanos y difusos.

El método propuesto consiste en resolver los dos grandes problemas que se han encontrado en los trabajos anteriormente mencionados, los cuales son la escasez de datos y la evolución del gusto de los usuarios. Por ello, se propone un nuevo sistema de recomendación que utilice SVD para resolver el problema de escasez y desequilibrio de datos, y a su vez utilizar RNR para adaptar las recomendaciones y predicciones a los gustos recientes del usuario. Además de ello, se realiza una comparación del método propuesto con métodos de línea base como KNN y FM con el objetivo demostrar la potencialidad de las RNR en el ámbito de los sistemas de recomendación.

III. Metodología

A. Conceptos previos

Antes de presentar la metodología para nuestro SR, es necesario señalar la importancia de las redes neuronales recurrentes en el método propuesto. Pensar en el filtrado colaborativo como un problema de secuencias, permite que los datos sean tratados de la misma manera.

Lo que hace que una secuencia sea especial, es su dependencia y su orden. Para ello, la mejor forma de trabajar con secuencias es utilizando las redes neuronales recurrentes (RNR). Para poder entender cómo funcionan estas redes, en la figura 1 se presenta una comparación de las RNR y las redes feed-forward.

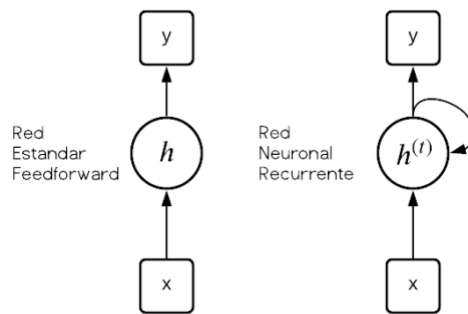


Figura 1. Arquitectura de una red neuronal feedforward y red neuronal recurrente. Imagen adaptada de (Sebastian Raschka, 2019)

Como podemos notar en la figura 1, las RNR se caracterizan por tener una retroalimentación en su neurona, haciendo que las secuencias se lleven de la mejor manera con este tipo de arquitectura y así poder tratar las dependencias de las secuencias. Para ello, estas RNR pueden ser de dos tipos según (Sebastian Raschka, 2019): 1) *Many-to-one* en donde a la capa de entrada se le pasa una secuencia y como salida devuelve un valor y 2) *Many-to-many* donde tanto la capa de entrada como de salida son una secuencia.

Una de las ventajas de las RNR es que estas permiten realizar entrenamiento tomando en cuenta eventos pasados y así predecir eventos que podrían pasar en un futuro, a este tipo de aprendizaje se lo conoce como basado en gradientes y retropropagación (*Backpropagation*) según lo explica (Sebastian Raschka, 2019) . El problema de este aprendizaje es que necesitamos ventanas de

tiempo muy grandes, generando que los pesos de las neuronas puedan explotar o desaparecer haciendo inestable nuestro modelo RNR, a este problema se lo como *Exploding/Vanishing*.

Para resolver este problema de desvanecimiento y explotación, se tienen dos nuevos tipos de redes neuronales recurrentes:

- *Long Shot Term Memory* (LSTM): Este tipo de RNR permite recordar estados previos de una secuencia, ya que estas cuentan con una unidad de memoria. Con ello, se puede utilizar dicha información para predecir cual será el siguiente valor y así evitar las grandes ventanas de tiempo como lo explica (Sebastian Raschka, 2019) .
- *Gated Recurrent Units* (GRU): Las redes GRU son un nuevo enfoque de RNR, cuyo concepto es similar a las LSTM. Este tipo de redes según (Sebastian Raschka, 2019) “tienen una arquitectura más simple que las LSTM; por lo tanto, son computacionalmente más eficientes en algunas tareas, pero menos potentes”.

Para el método propuesto se usó LSTM con el fin de resolver los problemas de desvanecimiento y explotación. Para ello en la figura 2 se explica cómo se forma el conjunto de entrada de datos para la capa de entrada LSTM del método propuesto.

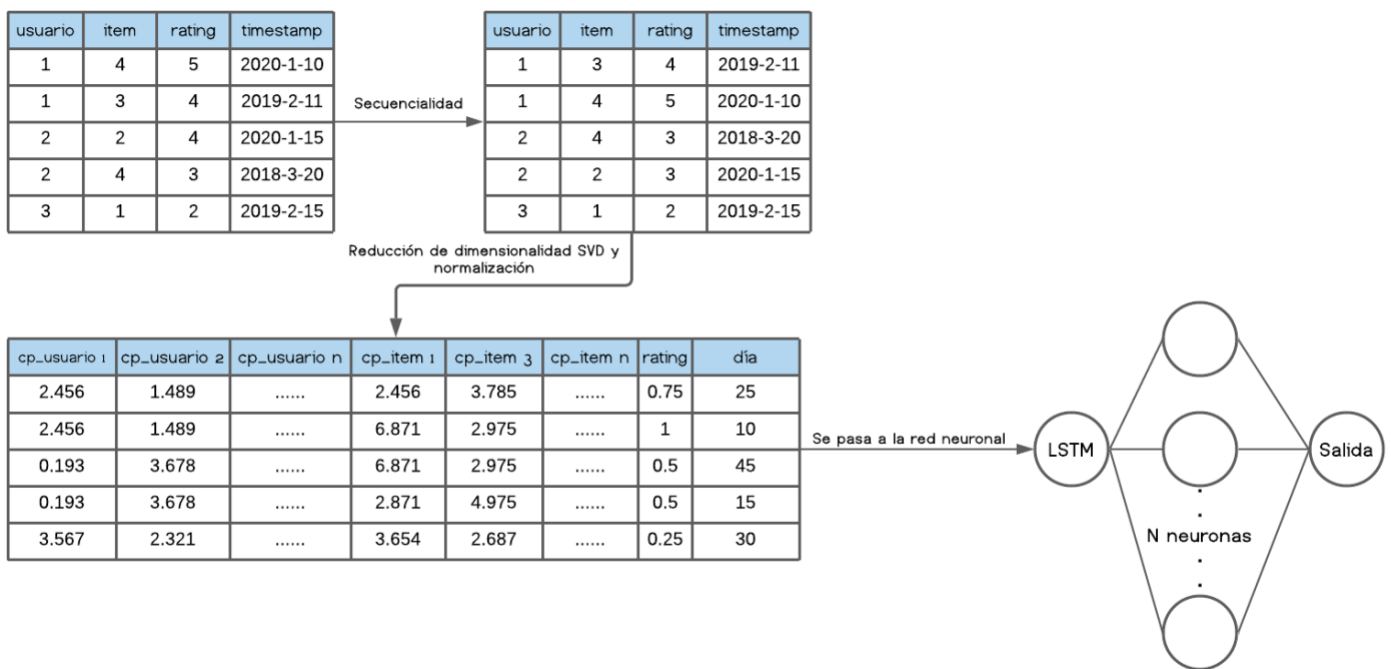


Figura 2. Proceso del método propuesto para combinar el filtrado colaborativo con la capa de entrada LSTM

Como se puede apreciar en la figura 2, es muy importante recalcar que para poder pasar el conjunto de datos a la capa de entrada LSTM se realiza una reducción de dimensionalidad mediante SVD, denotando la importancia de esta en el método propuesto. Con ello, la red neuronal recurrente con capa de entrada LSTM recibe el conjunto de datos mostrado en la figura 2, en donde podemos ver que el tiempo predomina para cada usuario.

Por otro lado, también hay que tomar en cuenta que para evaluar el entrenamiento de nuestro modelo utilizaremos la técnica de Validación Cruzada (*Cross-Validation*). La validación cruzada es un proceso que consiste en evaluar los resultados de un modelo con la finalidad de garantizar que dicho modelo es independiente de la partición que se obtenga de los datos (evitando un *overfitting* o sobreajuste); para ello utilizamos la técnica *KFolds*, que consiste en plegar nuestros datos k veces, con el fin de utilizar diferentes partes de nuestro dataset como entrenamiento y prueba, y así cubrir todo el conjunto de datos.

Por otro lado, el proceso *de optimización* consiste en reducir el error del modelo con el objetivo de obtener el Error (*loss*) más bajo, este proceso va de la mano con la técnica de *Fine-Tuning*, el cual permite buscar posibles errores y combinaciones de valores de parámetros (específicamente, de los hiperparámetros) que puedan mejorar el modelo. Dentro del proceso de *Fine-Tuning* se utiliza la técnica *GridSearchCV* la cual es una técnica organizada, exhaustiva y sistemática de probar todos los parámetros que le mandemos a nuestro modelo. Esta técnica define grupos de parámetros que serán probadas en todas sus combinaciones (un grupo a la vez). El objetivo es encontrar el conjunto de hiperparámetros óptimo. A continuación, se da una explicación más detallada de todos los conceptos revisados previamente, con el objetivo de entender de mejor manera el método propuesto.

B. Diseño del método

El método propuesto en este trabajo consiste en 10 pasos los cuales se pueden apreciar en la figura 3. De todos los pasos indicados en la figura 3, el método consta de 6 fases importantes: 1) Data gathering (extracción de datos), 2) Data wrangling (proceso de limpieza y transformación de datos), 3) Diseño y optimización, 4) Entrenamiento del modelo, 5) Predicciones y recomendaciones y 6) Evaluación de las predicciones y recomendaciones. Los pasos del método propuesto se explican a continuación.

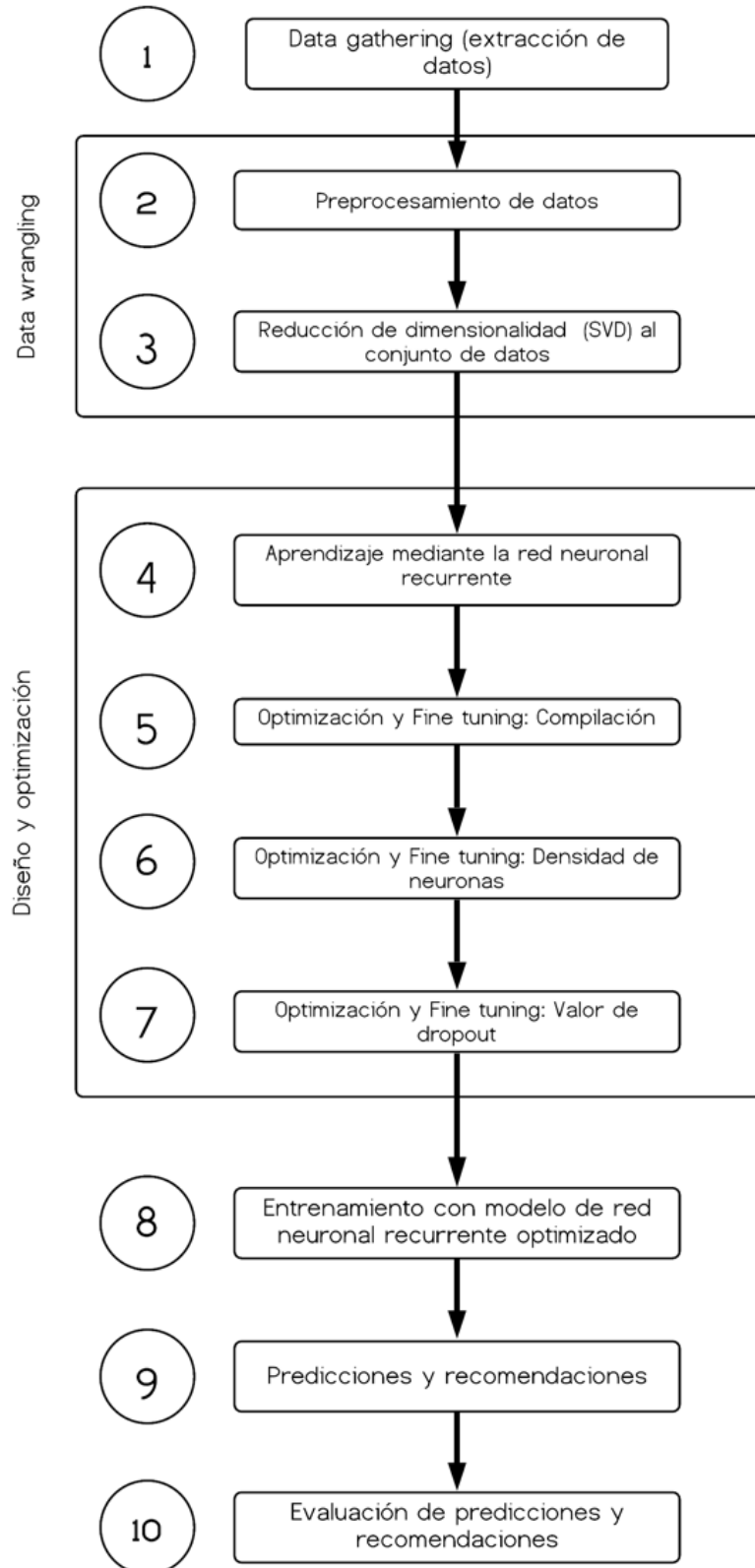


Figura 3. Arquitectura del método propuesto

- **Fase 1: Data gathering (extracción de datos)**

En esta fase se cumple el paso 1 indicado en la figura 3, el cual consiste en extraer los datos. Para ello se extrae el tiempo como característica fundamental ya que es de suma importancia tenerlo en nuestro dataset para poder dar secuencialidad a nuestro conjunto de datos. Además de ello, se extraerán los datos de usuarios e ítems que se van a recomendar, así como también la valoración del usuario hacia dicho ítem.

- **Fase 2: Data wrangling**

En esta fase se cumplen los pasos 2 y 3 que se visualizan en la figura 3: 2) Preprocesamiento y 3) Reducción de dimensionalidad usando SVD al conjunto de datos. Estos pasos se explican a continuación.

- *Paso 2 - Preprocesamiento:* En este paso lo que se busca es realizar una limpieza al conjunto de datos como: Ordenar al conjunto de datos acorde al tiempo y usuario, normalizar los valores de salida en rangos [0,1]. Con ello, se logra tener un conjunto de datos normalizado, estructurado y con un orden acorde al tiempo.
- *Paso 3 - Reducción de dimensionalidad:* Con el conjunto de datos pre procesado se procede a realizar la reducción de dimensionalidad aplicando SVD con el objetivo de mitigar el sesgo de datos. Para ello, se obtienen los componentes para usuarios y también para ítems. Luego de ello, se realiza una mezcla de ambos componentes en el conjunto de datos normalizado logrando mitigar el sesgo del dataset con los componentes obtenidos para cada característica descrita.

- **Fase 3: Diseño y optimización**

Esta fase es muy importante puesto que aquí se realiza el diseño de un modelo inicial de red neuronal recurrente, la cual pasará por un proceso de optimización y *Fine-Tuning* que está compuesto por tres etapas: 1) Compilación, 2) densidad de neuronas y 3) Regularización de dropout. Esto nos da como resultado un nuevo y mejorado modelo de red neuronal recurrente. En dicha fase se cumplen los pasos 4, 5, 6 y 7 de la figura 3. Estos se explican a continuación.

- *Paso 4 - Aprendizaje mediante la red neuronal recurrente:* En este paso se diseña un modelo inicial de red neuronal recurrente la cual es entrenada con el objetivo de

realizar una evaluación utilizando *Cross-Validation* con kFolds y así tener un valor de referencia a mejorar en nuestro proceso de optimización y fine-tuning.

- *Paso 5 – Optimización y Fine-Tuning Compilación:* En esta etapa se busca obtener los mejores parámetros correspondientes a los valores de: El número de muestras que se propagan a través de la red (tamaño del batch), el número de veces que se ejecutará el modelo con los datos de entrenamiento (epochs) y el optimizador del modelo de red neuronal.
- *Paso 6 – Optimización y Fine-Tuning Densidad de neuronas:* Durante esta etapa se busca el mejor número de neuronas para cada una de las capas de nuestro modelo de red neuronal.
- *Paso 7 – Optimización y Fine-Tuning Valor de dropout:* Como última etapa realizamos la regularización de dropout con la finalidad de encontrar el mejor valor para cada capa. La capa de dropout permite apagar y encender neuronas con el objetivo de que estas no sean tan dependientes de sus pesos evitando así el sobreentrenamiento.

- ***Fase 4: Entrenamiento con el modelo de red neuronal recurrente optimizado***

Una vez obtenido el modelo de red neuronal recurrente optimizado, se llega a esta fase, la cual es el paso 8 según la figura 3. En ese paso se procede a realizar el proceso de entrenamiento con el modelo RNR optimizado y de la misma manera evaluar el mismo utilizando *Cross-Validation* con KFolds con el objetivo de comparar el resultado del modelo optimizado contra el modelo inicial.

- ***Fase 5: Predicciones y recomendaciones***

Con el modelo RNR optimizado y evaluado, procedemos a realizar esta fase, identificado como el paso 9 en la figura 3. Con nuestro modelo entrenado, realizamos las predicciones con el conjunto de prueba y las recomendaciones a partir de nuevos conjuntos de datos. Con los resultados de predicción y recomendación se evaluará el SR mediante métricas de calidad.

- ***Fase 6: Evaluación de predicciones y recomendaciones***

Con las predicciones y recomendaciones realizadas en la fase anterior, se procede a evaluar dichos resultados con el objetivo de conocer la calidad de nuestro SR. Para la evaluación se utilizan las métricas de calidad: MAE para calcular el error de nuestro sistema, así como

también la métrica *Precision Score* que permite calcular la precisión del sistema en las recomendaciones que genera nuestro SR.

IV. Diseño de experimentos

A. Experimentos establecidos

Para el desarrollo del método propuesto, utilizamos el dataset público Movielens 1M. Este conjunto de datos está compuesto por usuarios que realizan una valoración hacia ciertas películas en un tiempo dado, donde los usuarios y películas están representados por un identificador numérico. Movielens es un conjunto de datos adecuado ya que este contiene el tiempo en el que un usuario realizó la valoración hacia una película. Esto va a permitir dar secuencialidad al conjunto de datos. En la tabla 1 se encuentra información referente al contenido de Movielens.

Tabla 1. Principales propiedades del conjunto de datos usado en el experimento

Dataset	Total de datos	Número de ítems	Número de usuarios	Rango de valoración
Movielens	1,000,209	3706	6040	Escala de 5 estrellas

Con la información obtenida en la tabla 1, se realiza un preprocesamiento con el objetivo de normalizar los datos. Realizamos una normalización al rango de valoración el cual será el valor de salida de nuestro modelo. Para ello normalizamos dicho rango en valores de $[0,1]$ puesto que las redes neuronales tienden a ser muy sensibles con datos no normalizados. Por otro lado, re-indexamos a los usuarios e ítems para evitar códigos faltantes y tener una secuencialidad en los mismos.

De igual forma, se ordena al conjunto del dataset acorde al tiempo (número del día acorde a los 365 del año) en el que el usuario realizó una votación, con el fin de dar secuencialidad al conjunto de datos. Posterior a ello, se utiliza SVD para reducir la dimensionalidad del conjunto de datos y aliviar la dispersión. Se generan dos matrices transpuestas (usuarios-ítems) e (ítems-usuarios) para luego obtener los componentes de cada matriz y traspasar dichos componentes obtenidos hacia el conjunto de datos original. Esto da como resultado un conjunto de datos compuesto por: n componentes de usuario, n componentes de ítems, ratings y tiempo. Por último, se divide el conjunto de datos en entrenamiento y prueba sin perder la secuencialidad

de datos impuesta por el tiempo. En la tabla 2, podemos observar la división planteada, así como el número de componentes que se probaron en SVD.

Tabla 2. Principales parámetros para el método propuesto

Dataset	Número de datos de entrenamiento	Número de datos de prueba	Número de componentes SVD
Movielens	800,167	200,041	10,20,40,100

a. Arquitectura de los modelos de red neuronal

Una vez realizado la fase de preprocesamiento, se realiza el diseño del modelo inicial de red neuronal. Con el objetivo de comprar modelos de red neuronal y a su vez enriquecer nuestros experimentos, se ha planteado dos diseños de red neuronal: Una red neuronal estándar (RNE) y una red neuronal recurrente (RNR). El objetivo de ello es demostrar la potencialidad de las redes RNR al trabajar con series temporales. En la tabla 3 se muestra las topologías propuestas.

Tabla 3. Topología de los modelos de red neuronal

Modelo	Topología	Parámetros de compilación		
		Epochs	Tamaño del batch	optimizador
RNR	1 capa LSTM con 16 neuronas con función de activación relu	5	1024	adam
	1 capa de dropout con valor de 0.3			
RNE	1 capa de salida con 1 neurona con función de activación linear	5	1024	adam
	1 capa con 16 neuronas			
	1 capa de salida con 1 neurona			

Como se puede apreciar en la tabla 3, se tienen dos modelos de red neuronal. Una red neuronal recurrente que presenta una capa LSTM que permite trabajar con datos secuenciales, y una capa de dropout. Por otro lado, tenemos una red neuronal estándar que solamente contiene una capa estándar. Ambos modelos cuentan con una capa de salida con

1 neurona debido a que ambos retornan una predicción acerca de la valoración del usuario hacia un ítem.

Con los modelos iniciales diseñados, se realiza una evaluación utilizando la técnica de *Cross-Validation* con KFold. Para este proceso se ha utilizado un kFolds con K=3 utilizando la métrica MAE como función de pérdida, con el objetivo de crear tres conjuntos de datos y evitar la fluctuación de los resultados. El resultado de esta evaluación será la base que ha ser mejorada en el proceso de optimización y *Fine-Tuning*.

b. Proceso de optimización y fine-tuning

Con el diseño inicial de RNR ya creado, se procede a realizar el proceso de optimización y *Fine-Tuning*, con el objetivo de mejorar el modelo inicial propuesto en la tabla 3. Este proceso consta de tres fases como se lo explico en la sección de metodología. Para la selección de los mejores parámetros se utiliza la técnica *GridSearchCV*, la cual permite realizar una prueba exhaustiva con todas las combinaciones de los parámetros a fin de encontrar el mejor grupo de valores. A continuación, se explica cada una de las fases.

- ***Fase 1: Compilación***

En esta fase se procede a realizar una combinación de valores para los parámetros de compilación de la tabla 3. Lo que se busca en esta fase es encontrar los mejores valores para dichos parámetros. En la tabla 4 se muestran los valores probados en esta fase.

Tabla 4. Parámetros de prueba para la fase de compilación

Tamaños del batch	Epochs	Optimizadores
64, 128 ,256, 512, 1024	5, 10, 20, 50	adam. rmsprop

- *Fase 2: Densidad de Neuronas*

En la fase de densidad de neuronas se procede a optimizar el número de neuronas para cada capa del modelo de red neuronal recurrente según la tabla 3. La capa que no se optimiza en este caso es la capa de salida puesto que dicha capa solamente deberá devolver un valor, por lo que solamente se debe tener 1 neurona. En la tabla 5 se aprecia la cantidad de neuronas probadas.

Tabla 5. Cantidad de neuronas para la fase de Densidad de Neuronas

Cantidad de neuronas
16, 60, 128, 256, 512

- *Fase 3: Regularización de dropout*

Con la regularización de dropout se busca encontrar el mejor valor para dicha capa, con el objetivo de que las neuronas no se vuelvan tan dependientes de sus pesos y así lograr evitar el sobre entrenamiento. En la tabla 6 se muestra la cantidad de pruebas que se realizaron para obtener el mejor dropout.

Tabla 6. Valores de dropout probados en la fase de regularización de dropout

Valores de dropout
0.1, 0.2, 0.3, 0.4, 0.5

Con la optimización realizada, se procede a realizar el aprendizaje del nuevo modelo de red neuronal recurrente optimizado con los parámetros obtenidos por el proceso de optimización y *Fine-Tuning*. Con el entrenamiento del nuevo modelo, se procede a realizar la evaluación de este mediante *Cross-Validation* con kFolds. Por último, realizamos las predicciones y recomendaciones utilizando este nuevo modelo optimizado para luego evaluar dichos resultados utilizando las métricas de calidad.

B. Medidas de calidad

Con la finalidad de medir la calidad de nuestro SR, utilizamos métricas de calidad las cuales permiten medir el error y la precisión de nuestro método. Antes de utilizar estas métricas se debe realizar una conversión de los valores de salida que fueron normalizados al inicio del preprocesamiento. Con los valores transformados a la escala original se procede a evaluar nuestro SR. Para este trabajo se ha decidido utilizar la métrica MAE (Mean Absolute Error) que según (Hurtado Ortiz, 2020) mide el error de las recomendaciones realizadas entre el conjunto de test y el conjunto de predicción.

$$MAE = \frac{\sum_{i=1}^n |y_i - y_{pred_i}|}{n}$$

Donde:

- y_i : Valoración real del usuario hacia el item
- y_{pred_i} : Predicción de la valoración del usuario hacia el item
- n : Conjunto de datos de prueba

Por otro lado, para medir la precisión de nuestro sistema de recomendación utilizamos la métrica *Precision Score* el cual según (Jesús Bobadilla et al., 2020) y (Hurtado Ortiz, 2020) “indica la proporción de ítems recomendados relevantes del número total de ítems recomendados”. Para utilizar dicha métrica transformamos los resultados de predicción a valores binarios (0,1) a partir de un umbral. Dicho umbral funciona de la siguiente manera: Aquellos valores que tengan una valoración mayor o igual a 4 se convierten en 1, caso contrario se convierten en 0. Dicha transformación, se aplica tanto para los valores reales, como a los valores de predicción. Con esta conversión, se podrá utilizar esta métrica para evaluar las recomendaciones que da nuestro SR.

$$precision = \frac{verdaderos\ positivos}{verdaderos\ positivos + falsos\ positivos}$$

Donde:

- *verdaderos positivos* : Valor donde la valoración real y la predicción son 1.
- *falsos positivos*: Valor donde la valoración real es 0 y la predicción es 1.

Con la finalidad de enriquecer aún más nuestros experimentos, se ha comparado los dos modelos de red neuronal RNE y RNR, realizando el mismo modelo, pero con la diferencia de

que en el preprocesamiento no se realiza reducción de dimensionalidad con SVD. Todos estos métodos serán evaluados y comparados con el método propuesto en este trabajo.

V. Resultados y discusión

En esta sección presentamos los resultados de los experimentos realizados. Con respecto a la reducción de dimensionalidad, generalmente se suelen utilizar de 10 a 40 componentes en SVD, con esto en consideración se ha optado por utilizar 10 componentes ya que esta dentro del rango usual y además de ello, el tiempo de procesamiento es ideal como también sus resultados. Con el número de componentes escogido y los modelos de red neuronal diseñados, se procede a evaluarlos con la finalidad de tener un resultado a ser mejorado en la optimización. El resultado obtenido en dicha evaluación con *Cross-Validation* y kFolds es 0.29 con la métrica MAE. En base a ello, en la fase de optimización y fine-tuning se obtuvieron los siguientes resultados que se aprecian en la tabla 7.

Tabla 7. Mejores parámetros obtenidos en la fase de optimización y fine-tuning

Tamaño del batch	Epochs	Optimizador	Cantidad de neuronas	Dropout
1024	10	Rmsprop	Capa 1 = 16	0.4

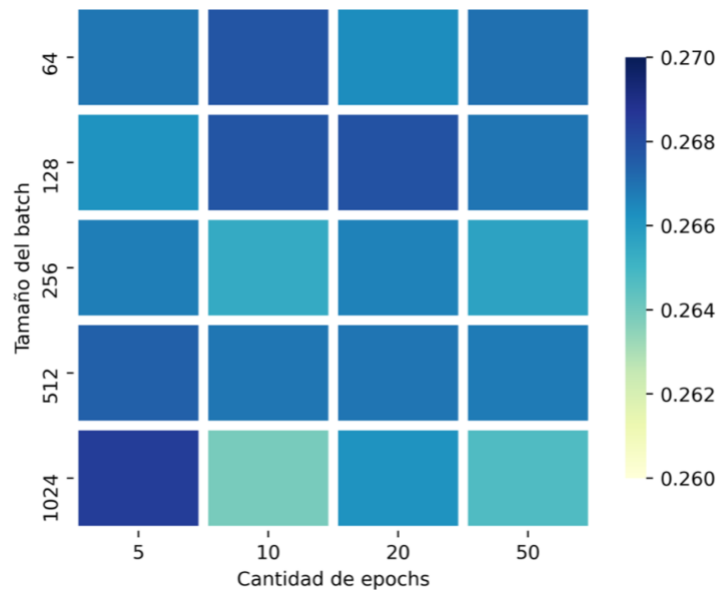


Figura 4. Fase de compilación: Prueba de todos los grupos de parámetros evaluados en el modelo RNR usando el optimizador rmsprop

En la figura 4 podemos apreciar lo descrito en la tabla 7. En la figura 4 la cual está representada como un mapa de calor, podemos notar que los mejores parámetros están compuestos por un tamaño de batch de 1024 y una cantidad de 10 epochs utilizando el optimizador rmsprop, obteniendo un valor de 0.26, mejorando el valor de error inicial. De la misma manera se puede visualizar como un tamaño de batch pequeño presenta un mayor error sin importar la cantidad de epochs que se tenga. Sin embargo, podemos notar claramente que el peor resultado está dado por un tamaño de batch de 1024 y una cantidad de 5 epochs, dando a entender que utilizar los extremos de los parámetros evaluados, dan malos resultados, pero siguen siendo más bajos que el error inicial, lo cual nos permite intuir que el optimizador rmsprop trabaja muy bien con redes neuronales recurrentes.

Con la optimización realizada ya se obtiene el nuevo modelo RNR optimizado, por lo cual se procede a realizar el entrenamiento de este utilizando los parámetros descritos en la tabla 7. Dicho entrenamiento se presenta en la figura 5.

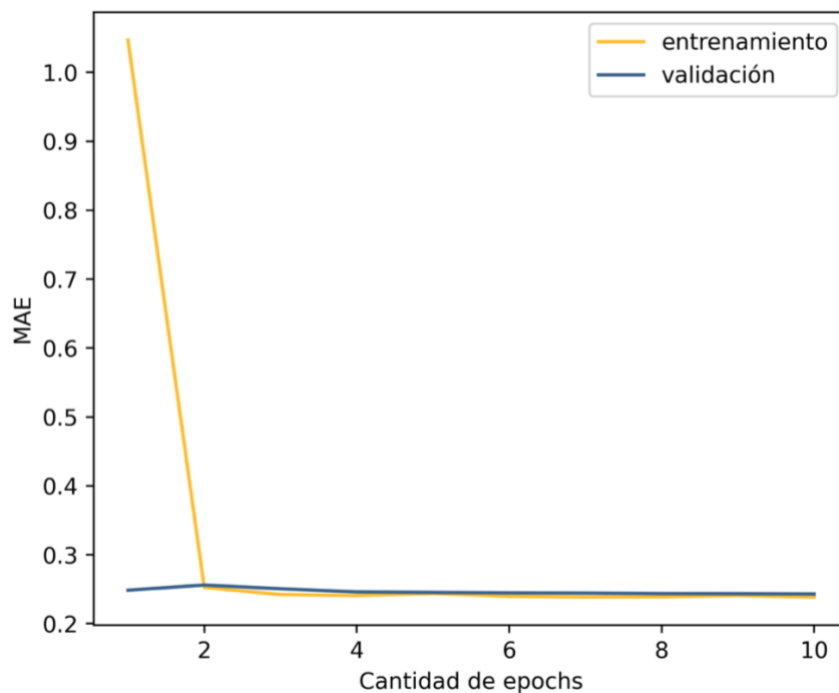


Figura 5. Entrenamiento y validación del método propuesto

Como podemos apreciar en la figura 5, el valor entrenamiento es muy alto al inicio, pero a medida que se va avanzando en la cantidad de epochs el error tiende a bajar. De igual manera sucede en la validación, con la diferencia de que en el primer epoch no existe un error tan alto. Sin embargo, tanto en entrenamiento como en validación se tiene una tendencia a bajar de tal

manera que en el último epoch se tiene el error más bajo tanto en entrenamiento como en validación, denotando a su vez la importancia de la optimización.

Por último, se realiza la evaluación del método propuesto utilizando las métricas de calidad descritas en la sección anterior las cuales son MAE y *Precision Score*. Muy importante tomar en cuenta, que esta evaluación se realiza escalando los resultados del SR a sus valores originales, puesto que todos los resultados mostrados anteriormente han sido obtenidos con los valores normalizados. Con esto en consideración, el error tiende a subir ya que se escalan los valores normalizados a sus escalas originales y esto nos permite medir de manera real nuestro SR.

Por otro lado, se compara el método propuesto el cual combina reducción de dimensionalidad y redes neuronales recurrentes al cual lo vamos a denominar (RNRRD), con modelos diseñados y evaluados en la fase de experimentación. Para dicha comparación, se genera un modelo (RNERD) que combina la red neuronal estándar diseñado en la tabla 3 con la reducción de dimensionalidad, también presentamos un modelo denominado RNR ya que solamente utiliza la red neuronal recurrente sin haber realizado una reducción previa, y por último tenemos una RNE la cual es una red neuronal estándar sin utilizar reducción de dimensionalidad. Con ello, en la tabla 8 se presentan los resultados obtenidos para cada uno de estos modelos.

Tabla 8. Comparación de modelos de red neuronal utilizando las métricas de calidad

Modelo	MAE	Precision score
RNE	0.8693	0.5827
RNERD	0.7851	0.7658
RNR	0.8019	0.7522
RNRRD	0.7155	0.8151

Como se puede apreciar en la tabla 8, podemos notar como la RNR es mejor que la RNE en primera instancia, dando mejores resultados tanto en MAE como en *Precision Score*. Sin embargo, al utilizar reducción de dimensionalidad podemos ver como la RNERD es mejor que la RNR, dando a entender la importancia de la reducción de dimensionalidad en los SR. Por último, podemos notar como el método propuesto RNRRD, es mejor que RNRED en ambas métricas y además de ello es el que menor error y mejor *Precision Score* tiene de todos los

modelos descritos en la tabla 8. Con ello podemos notar la potencialidad del método propuesto. Por ello, se compara nuestro RNRRD con métodos de línea base tales como KNN, FM y sus variantes. En la tabla 9 se presenta una comparación de todos estos métodos.

Tabla 9. Comparación del método propuesto con líneas de base

Métodos	MAE
RNRRD	0.7155
KNN JMSD (Jesús Bobadilla et al., 2012)	0.7421
NMF (Lee & Seung, 1999)	1.2412
MF (Salakhutdinov & Mnih, 2009)	0.6842
BNMF (Hernando et al., 2016)	0.6613

Como se puede notar en la tabla 9, los métodos de factorización matricial como MF y BNMF son buenos y presentan los mejores resultados, sin embargo, nuestro método compite muy bien con estos modelos. Por otro lado, podemos notar que nuestro RNRRD es mejor que el método NMF y el KNN JMSD. Con ello, podemos notar la gran potencialidad de las redes neuronales recurrentes en los SR y abre puerta al diseño de nuevos modelos basados en nuestro enfoque. Además de ello, tomemos en cuenta que el método propuesto se adapta a la evolución de los gustos y preferencias de los usuarios mientras que los métodos de línea base no lo hacen, dándole otra ventaja a nuestro método.

VI. Conclusiones

Los resultados presentados en este trabajo confirman nuestra hipótesis: La reducción de dimensionalidad es muy importante en los SR para reducir errores y mejorar precisión, como también las redes neuronales recurrentes trabajan muy bien con datos secuenciales como las series temporales. De igual manera vemos como el método propuesto compite muy bien con métodos del estado del arte que han dado muy buenos resultados. El trabajo realizado da a conocer una nueva metodología con respecto a las nuevas técnicas que se pueden utilizar para mejorar los sistemas de recomendación. Este trabajo se lo deja como una línea base para futuras investigaciones con la finalidad de que se puedan aplicar nuevas técnicas tales combinar conceptos de PLN (Procesamiento de Lenguaje Natural) con redes neuronales recurrentes para recomendar un siguiente ítem acorde a los ítems consumidos por el usuario. Como también aplicar otras técnicas de reducción de dimensionalidad tales como PCA (Análisis de

componentes principales) que consiste en relacionar las características del conjunto de datos con el fin de obtener un conjunto de características nuevas llamadas componentes principales. Así mismo, combinar el método propuesto con otras técnicas con el objetivo mejorar los resultados descritos en este trabajo.

VII. Referencias

- Aleksandrova, M., Brun, A., Boyer, A., & Chertov, O. (2017). Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem. *Journal of Intelligent Information Systems*, 48(2), 365–397. <https://doi.org/10.1007/s10844-016-0418-3>
- Bobadilla, J., Ortega, F., Hernando, A., & Arroyo, A. (2014). A Balanced Memory-Based Collaborative Filtering Similarity Measure. *International Journal of Intelligent Systems*, 29(2), 495–524. <https://doi.org/10.1002/int.21556>
- Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems*, 46, 109–132. <https://doi.org/10.1016/j.knosys.2013.03.012>
- Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, 23(6), 520–528. <https://doi.org/10.1016/j.knosys.2010.03.009>
- Bobadilla, Jesús, Gutiérrez, A., Alonso, S., & Hurtado, R. (2020). A Collaborative Filtering Probabilistic Approach for Recommendation to Large Homogeneous and Automatically Detected Groups. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6(2), 11. <https://doi.org/10.9781/ijimai.2020.03.002>
- Bobadilla, Jesús, Hernando, A., Ortega, F., & Gutiérrez, A. (2012). Collaborative filtering based on significances. *Information Sciences*, 185(1), 1–17. <https://doi.org/10.1016/j.ins.2011.09.014>
- Catalán, C. E. (n.d.). *SERIES TEMPORALES*.
- Christakou, C., Vrettos, S., & Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, 16(5), 771–792. <https://doi.org/10.1142/S0218213007003540>
- Cruz, B., Martínez, S., Abed, R., Abalo, G., Lorenzo, G., Matilde, M., Cruz, I. B., Martínez, S. S., Abed, A. R., Abalo, G., & Lorenzo, M. G. (2007). *Redes neuronales recurrentes para el análisis de secuencias Recurrent neural network for sequences analysis*.
- Devooght, R., & Bersini, H. (2016). *Collaborative Filtering with Recurrent Neural Networks*. <http://arxiv.org/abs/1608.07400>
- Guan, X., Li, C. T., & Guan, Y. (2017). Matrix Factorization with Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems. *IEEE Access*, 5, 27668–27678. <https://doi.org/10.1109/ACCESS.2017.2772226>
- Hernando, A., Bobadilla, J., & Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems*, 97, 188–202. <https://doi.org/10.1016/j.knosys.2015.12.018>
- Hurtado Ortiz, R. (2020). *Recomendación a grupos de usuarios usando el concepto de singularidades*. http://oa.upm.es/58148/1/REMIGIO_ISMAEL_HURTADO_ORTIZ.pdf

- Hurtado, R., Bobadilla, J., Bojorque, R., Ortega, F., & Li, X. (2019). A new recommendation approach based on probabilistic soft clustering methods: A scientific documentation case study. *IEEE Access*, 7, 7522–7534. <https://doi.org/10.1109/ACCESS.2018.2890079>
- Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788–791. <https://doi.org/10.1038/44565>
- Liu, H., Hu, Z., Mian, A., Tian, H., & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. *Knowledge-Based Systems*, 56, 156–166. <https://doi.org/10.1016/j.knosys.2013.11.006>
- McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. *Conference on Human Factors in Computing Systems - Proceedings*, 1097–1101. <https://doi.org/10.1145/1125451.1125659>
- Moreno, B. D. V., Ortiz, R. I. H., & Rivera, D. A. M. (2019). A new approach hybrid recommender system of item bundles for group of users. *2019 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2019, Ropec*. <https://doi.org/10.1109/ROPEC48299.2019.9057121>
- Said, A., & Bellogín, A. (2014). Comparative recommender system evaluation: Benchmarking recommendation frameworks. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 129–136. <https://doi.org/10.1145/2645710.2645746>
- Salakhutdinov, R., & Mnih, A. (2009). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 1–8.
- Sebastian Raschka, V. M. (2019). Python Machine Learning. In *Taiwan Review* (Second edi, Vol. 69, Issue 4). Packt Publishing Ltd.
- Wu, M. L., Chang, C. H., & Liu, R. Z. (2014). Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. *Expert Systems with Applications*, 41(6), 2754–2761. <https://doi.org/10.1016/j.eswa.2013.10.008>
- Yang, Z., Wu, B., Zheng, K., Wang, X., & Lei, L. (2016). A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access*, 4(c), 3273–3287. <https://doi.org/10.1109/ACCESS.2016.2573314>
- Yu, J., Gao, M., Rong, W., Song, Y., & Xiong, Q. (2017). A Social Recommender Based on Factorization and Distance Metric Learning. *IEEE Access*, 5, 21557–21566. <https://doi.org/10.1109/ACCESS.2017.2762459>
- Zhu, B. O., & Hurtado, R. (2018). *An Efficient Recommender System Method Based on the Numerical Relevances and the Non-Numerical Structures of the Ratings*. 49935–49954. <https://doi.org/10.1109/ACCESS.2018.2868464>