

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



TRABAJO FIN DE MÁSTER

**Estudio y análisis del uso de redes
siamesas en estrategias de
recomendación basadas en
contenido y de filtrado
colaborativo**

Máster Universitario en Ciencia de Datos

**Autor: Serrano Salas, Nicolás
Tutor: Bellogín Kouki, Alejandro
Departamento de Ingeniería Informática**

febrero 2023

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© Febrero 2023 por UNIVERSIDAD AUTÓNOMA DE MADRID

Francisco Tomás y Valiente, nº 1

Madrid, 28049

Spain

Serrano Salas, Nicolás

Estudio y análisis del uso de redes siamesas en estrategias de recomendación basadas en contenido y de filtrado colaborativo

Serrano Salas, Nicolás

Francisco Tomás y Valiente, nº 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

PREFACIO

Este Trabajo de Fin de Máster ha sido diseñado con el objetivo de la investigación del uso de Sistemas de Recomendación junto con las redes siamesas. Su propósito principal es hacer un estudio previo de los actuales tipos de implementaciones, así como crear unos modelos del que partir si algún desarrollador quiere utilizarlos como sistema de recomendación o para la evaluación de sus propios sistemas.

Como desarrollador de este Trabajo de Fin de Máster espero que sea de utilidad para aquellas personas interesadas tanto en los sistemas de recomendación como en las redes siamesas.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a Alejandro Bellogín Kouki por la oportunidad de desarrollar este Trabajo de Fin de Máster, la orientación durante todo su proceso y ver la posibilidad de desarrollar un artículo de investigación con base en el propio TFM.

También quiero agradecer a mis amigos del máster y el grado, así como a los chavales del barrio y de Discord que están siempre conmigo todo este tiempo.

Por último, a mi familia, sobre todo a mi madre, que me ha permitido estudiar sin problemas, preocupándose por mí todo este tiempo.

RESUMEN

Los sistemas de recomendación se utilizan para mostrar información diversa y de interés al usuario. Existen diferentes estrategias, pero tanto en la recomendación basada en contenido y como en filtrado colaborativo, se utiliza la similitud bien entre los artículos o entre los usuarios. Así mismo, las redes siamesas se están utilizando hoy en día para capturar la similitud de artículos en el dominio de imágenes. El objetivo de este trabajo es estudiar el comportamiento de estas redes y sus diferentes arquitecturas en los diferentes problemas de recomendación, plantear cómo construir diferentes tipos de sistemas de recomendación con redes siamesas para distintos tipos de problema y evaluar el impacto que pueden tener según la estrategia de recomendación utilizada o el dominio de aplicación del que provengan los datos.

Para ello, se han estudiado todos los artículos del estado del arte que utilizaban sistemas de recomendación con redes siamesas y se han catalogado todos los tipos de aproximaciones en diferentes categorías. También, se han estudiado las implementaciones de aquellos artículos considerados más relevantes, creando una serie de implementaciones replicables para que cualquier lector pueda implementar un sistema de recomendación con redes siamesas. Después, basándonos en las implementaciones anteriores, se han estudiado tres problemas concretos y se han evaluado los resultados del sistema de recomendación en los mismos.

Finalmente, se ha puesto el código de los experimentos y los modelos a libre disposición para su reutilización por parte del lector.

PALABRAS CLAVE

Sistemas de Recomendación, Redes Siamesas, Estudio, Basado en Contenido, Filtrado Colaborativo, Arquitectura, Evaluación, Revisión

ABSTRACT

Recommender systems are used to display diverse and useful information to the users. There are different strategies but in both content-based and collaborative filtering strategies, similarity is used either between items or between users. At the same time, Siamese networks are being used to capture the similarity of articles in the image domain. The objective of this report is to study the behavior of these networks and their different architectures in the different recommendation problems, to propose how to build different types of recommendation systems with Siamese networks for different types of problems, and to evaluate the impact they have depending on the recommendation strategy used or the application domain from which the data come from.

With this goal in mind, we have surveyed every article from the state-of-the-art that used recommendation systems with Siamese networks, and we have cataloged them in different approaches. Also, we have studied the implementations of the most relevant articles, creating a series of replicable implementations so that any reader could implement a recommendation system with Siamese networks. Then, based on these implementations, we have studied three specific problems, and we have evaluated the results of the recommender systems in those problems.

Finally, the experimental code and models has been left for free reuse by the readers.

KEYWORDS

Recommendation System, Siamese Network, Review, Content-based, Collaborative Filtering, Architecture, Evaluation, Review

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Propuesta	1
1.3	Estructura	2
2	Estado del arte	3
2.1	Sistemas de Recomendación	3
2.2	Redes Siamesas	5
2.3	Sistemas de Recomendación con Redes Siamesas	8
3	Implementación	15
3.1	Experimentos Basados en Contenido	17
3.2	Experimentos Filtrado Colaborativo	20
4	Experimentos y Resultados	23
4.1	1º experimento	23
4.2	2º experimento	25
4.3	3º experimento	26
5	Conclusiones y trabajo futuro	31
5.1	Conclusiones	31
5.2	Trabajo futuro	32
	Bibliografía	35
	Acrónimos	37
	Apéndices	39
A	Consultas bases de datos	41

LISTAS

Lista de figuras

2.1	Arquitectura de Red Siamesa	6
2.2	Arquitectura de Red Siamesa de Tripletas	8
2.3	Cambio de las distancias en una tripleta	11
3.1	Arquitectura de un problema de recomendación con redes siamesas	16
3.2	Arquitectura de la red convolucional propuesta para MNIST	19
4.1	Evaluación experimento 3, Triplet Network	29

Lista de tablas

2.1	Catálogo de artículos según dominio	9
2.2	Catálogo de artículos según uso de la red	11
2.3	Catálogo de artículos según red <i>feedforward</i>	12
2.4	Artículos junto a sus métricas de evaluación	14
3.1	Artículos con código	15
4.1	Especificaciones Hardware	23
4.2	Recall@10 Experimento 1	24
4.3	Precision@10 Experimento 1	25
4.4	Recall@10 Experimento 2	26
4.5	Precision@10 Experimento 2	26
4.6	Recall@K y Precision@K Experimento 3 Triplet Network	28
4.7	Recall@10 y Precision@10 dependiendo de vecindarios	28
4.8	Recall@K y Precision@K Experimento 3, Basado en Usuarios	29

INTRODUCCIÓN

En este primer capítulo se define cuál es la motivación y propuesta del proyecto, así como se detallan los objetivos principales de este Trabajo de Fin de Máster y la estructura de este documento.

1.1. Motivación del proyecto

Las **Redes Siamesas (SNN)** aparecieron en 1994 como arquitectura de red neuronal artificial, donde dos redes neuronales idénticas, en la época perceptrones, calculaban la similitud entre dos elementos. Aunque han sido olvidadas durante muchos años, gracias al avance en arquitecturas de redes neuronales artificiales, actualmente están siendo altamente utilizadas en el dominio multimedia, donde se puede aprovechar estos avances obteniendo una alta precisión en el cálculo de la similitud entre dos elementos.

Así mismo, los sistemas de recomendación toman un papel muy importante a la hora de mostrar información diversa y de interés al usuario. Existen diferentes estrategias a la hora de abarcar un problema de recomendación, las principales que define el libro de referencia de sistemas de recomendación [1] son la recomendación basada en contenido y el filtrado colaborativo. En ambas estrategias se utiliza la similitud bien entre los artículos o entre los usuarios.

1.2. Propuesta

En este trabajo se propone el uso de **SNN**, una arquitectura de modelos de red neuronal artificial que sirve para capturar la similitud entre diferentes entradas de la red. El objetivo es estudiar el comportamiento de estas redes y sus diferentes arquitecturas en los diferentes problemas de recomendación, así como el impacto que pueden tener según la estrategia de recomendación utilizada o el dominio de aplicación del que provengan los datos.

Los objetivos de este proyecto son:

- 1.– Conocer el estado del arte de las redes siamesas para recomendación.
- 2.– Plantear modelos de recomendación basados en redes siamesas, tanto para estrategias de filtrado colaborativo como basadas en contenido, dependiendo del dominio de los datos y la arquitectura de la red.
- 3.– Evaluar los modelos estudiados mediante medidas de ranking como Precision y Recall.

1.3. Estructura del documento

En el cuerpo de este Trabajo de Fin de Máster podemos encontrar en este primer capítulo que se ha introducido y detallado la propuesta y motivación del proyecto. En el siguiente capítulo (Capítulo 2) se presenta el estado del arte sobre sistemas de recomendación, las SNN y un estudio de las integraciones que han sido descubiertas en los últimos años. A continuación (Capítulo 3) se detalla la implementación de los diferentes sistemas de recomendación con redes siamesas para los experimentos. El siguiente capítulo (Capítulo 4) se detallan las configuraciones concretas utilizadas en los experimentos y los resultados obtenidos en los mismos. Finalmente, en el último capítulo (Capítulo 5) se incluyen las conclusiones y posibles trabajos futuros que pueden surgir a partir de este proyecto.

Como anexo A se incluyen una serie de consultas que han sido creadas para extraer los diferentes artículos de investigación que se han usado en la memoria a partir de diferentes bases de datos.

ESTADO DEL ARTE

En este segundo capítulo se presenta el estado del arte sobre sistemas de recomendación, las SNN y un estudio sobre las integraciones que han sido descubiertas en los últimos años.

2.1. Sistemas de Recomendación

Los sistemas de recomendación son una serie de herramientas y técnicas que proporcionan una sugerencia de artículo a un usuario. Esas sugerencias son importantes en procesos de toma de decisión como, qué me puede apetecer comprar, qué serie me puede apetecer ver, qué canción me puede gustar o qué sitio me puede gustar para comer.

Un artículo o *item* es, como se denota en los sistemas de recomendación, lo que es sugerido al usuario y un usuario es el sujeto de la recomendación. Los sistemas de recomendación normalmente se centran en un tipo de artículo, por ejemplo, películas, ropa o puestos de trabajo. De acuerdo a esto, su diseño, su interfaz gráfica de usuario y la técnica de recomendación utilizada es diferente en cada problema para proveer una sugerencia útil y eficiente para cada tipo de artículo.

Las recomendaciones suelen ser personalizadas, por lo que diferentes usuarios o grupos de usuarios reciben sugerencias diferentes, no obstante existen también las recomendaciones no personalizadas, siendo estas más fáciles de generar, como por ejemplo recomendar según la popularidad.

Las recomendaciones personalizadas se tienden a mostrar como un ranking de los artículos donde el sistema de recomendación intenta predecir cuál es el artículo más apropiado basándose en las preferencias del usuario, que pueden ser inferidas o expresadas por el mismo.

Existen diversas técnicas que se utilizan para implementar la funcionalidad de identificar que artículos son útiles para el usuario. En el manual de sistemas de recomendación [1] existen diferentes series de estrategias, pero este proyecto se centra en las estrategias de Basado en Contenido, Filtrado Colaborativo y Sistemas Híbridos, siendo esta última una combinación de las anteriores.

2.1.1. Basado en Contenido

En la estrategia de Basado en Contenido, el sistema aprende a recomendar artículos que son similares a los que al usuario le han gustado en un pasado. Las recomendaciones se crean considerando un perfil creado para el usuario, teniendo en cuenta las características de los artículos que le han gustado al mismo. Con ello, se busca la similitud frente a los atributos de los otros artículos para hacer la recomendación.

Por ejemplo, si el usuario solo ha escuchado *Für Elise (Para Elisa)* de Ludwig van Beethoven, el sistema puede aprender que ese tipo de música le gusta al usuario, pudiendo recomendar otra pieza del mismo compositor, la *Sonata para piano n.º 14 (Sonata Claro de luna)*; de género clásico como una pieza de Wolfgang Amadeus Mozart, *Sonata para piano n.º 16 (Sonata Facile)* o una pieza de piano más actual como *River Flows in You* de Yiruma. Sin embargo, si el usuario ha escuchado *Für Elise (Para Elisa)* de Ludwig van Beethoven, *Sonata para piano n.º 16 (Sonata Facile)* de Wolfgang Amadeus Mozart y *Nocturne op.9 No.2* de Chopin, el sistema seguramente nos recomiende otra pieza de música clásica de piano.

2.1.2. Filtrado Colaborativo

En la estrategia de Filtrado Colaborativo, la idea es recomendar al usuario artículos que le hayan gustado en un pasado a otros usuarios similares a él. Por lo tanto, dicha similitud es calculada mediante el historial de los usuarios. Es considerada la técnica más popular y ampliamente implementada entre los sistemas de recomendación ya que, entre otras cosas, no precisa de atributos o contenido de los ítems.

La respuesta que un usuario tiene sobre un artículo se divide en tres categorías: escalar, binaria y unaria. Las respuestas escalares se les conoce como *ratings*. Estas pueden ser numéricas, por ejemplo de 1 a 5 estrellas u ordinales, por ejemplo muy satisfecho, satisfecho, neutral, insatisfecho, muy insatisfecho. Las respuestas binarias solo toman dos posibles valores, siendo estos opuestos, por ejemplo me gusta / no me gusta, me interesa / no me interesa. Por último, las respuestas unarias capturan la interacción del usuario con el artículo, por ejemplo, si lo ha comprado o si lo ha visto, sin dar otra información explícita de la apreciación del usuario para dicho artículo. Normalmente, ya que los usuarios suelen interactuar con los artículos que les resultan interesantes, estas respuestas unarias proporcionan información de las preferencias del usuario.

Existen así mismo dos métodos en las que se dividen la estrategia de Filtrado Colaborativo: métodos de vecindarios y métodos basados en modelos.

Vecindarios

En el método basado en vecindarios, los ratings usuario-artículo se guardan en el sistema y se usan directamente para predecir el rating para nuevos artículos. Esto se hace de dos formas diferentes, basado en usuario y basado en artículo.

El método basado en usuario evalúa el interés de un usuario u para un artículo i usando los ratings de otros usuarios, llamados vecinos, para ese artículo. Esos vecinos tienen patrones de puntuación similar al usuario. Los vecinos del usuario u se denotan como v y, aunque hay varias formas de calcularlos, suelen ser aquellos cuyos ratings de los artículos comunes muestran una mayor correlación, de manera que tienen un cierto acuerdo a la hora de puntuar.

El método basado en artículo, por otro lado, evalúa el interés de un usuario u para un artículo i usando los ratings del propio usuario para artículos similares. El mecanismo para calcular artículos similares, al igual que pasa con los usuarios, puede variar, pero se basa en el histórico de puntuaciones entre dos artículos.

Modelos

En los métodos basados en modelos, en contraposición con los métodos basados en vecindarios que utilizan los ratings directamente en la predicción, usa los ratings para aprender un modelo predictivo. Estos modelos son entrenados con los datos disponibles y se utilizan para, más adelante, predecir ratings para los usuarios y artículos nuevos.

Algunos ejemplos de ello son la factorización de matrices como SVD [2] o las redes neuronales [3].

2.2. Redes Siamesas

La similitud es un concepto clave no solo en sistemas de recomendación sino en diferentes campos de la informática. Cuando se quiere medir la similitud entre dos elementos existen diferentes métodos: técnicas de comparación estadística, como la distancia euclídea o distancia coseno; técnicas centradas en la correlación, como el coeficiente de correlación de Pearson o el coeficiente de correlación de Spearman o técnicas basadas en métodos de aprendizaje automático.

Algunas de estas medidas, como los dos primeros tipos de técnicas, no son útiles cuando los dos elementos a comparar son listas de diferentes características, teniendo cada una diferentes significados o siendo de diferentes tipos de datos. Para solucionar esto, surgieron algunas técnicas basadas en métodos de aprendizaje automático. En 1993 se proponen las redes siamesas [4], donde se utilizan para la verificación de firma manuscrita.

Las redes siamesas es una arquitectura de red neuronal artificial [5] compuesta por varias redes *feedforward* idénticas (arquitectura y pesos), unidas en la salida. Se procesan a la vez los elementos los cuales se quieren comparar. Cada red procesa las características de uno de los elementos y compara las salidas mediante una métrica de distancia, como la distancia euclídea, determinando si son similares (para la distancia euclídea, valor cercano a 0) o diferentes (para la distancia euclídea, valor cercano a 1). En el entrenamiento, este resultado se compara con el etiquetado de los datos para determinar la eficacia del modelo mediante una función de pérdida.

Existen dos modelos principales de redes siamesas basándose en el número de parámetros de entrada a la red: parejas y tripletas.

2.2.1. Parejas

Las redes siamesas cuya entrada es una pareja de elementos son las primeras propuestas [4], a veces llamadas *Twin Neural Network*. Se emparejan pares de elementos y se aprende su similitud mediante una función de pérdida. Un diagrama de esta arquitectura se puede observar en la figura 2.1.

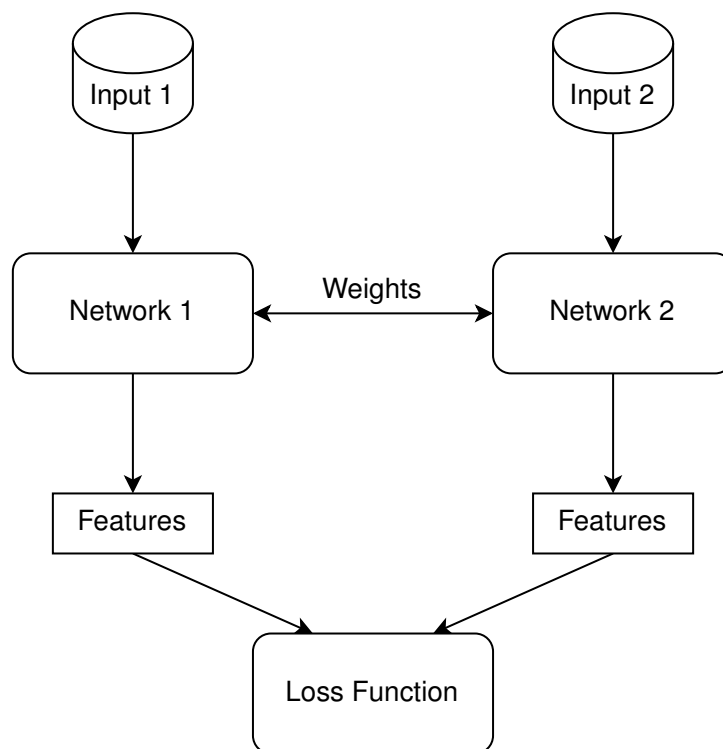


Figura 2.1: Arquitectura de Red siamesa de 2 entradas.

Existen dos funciones de pérdida que se usan en este tipo de modelos, la entropía binaria cruzada y la *contrastive loss*.

La entropía binaria cruzada es una buena opción para determinar si dos elementos son de la misma clase o de distintas clases, como se observa en la ecuación 2.1.

$$Loss = (Y)(-\log(Y_{pred})) + (1 - Y)(-\log(1 - Y_{pred})) \quad (2.1)$$

Descomponiendo la ecuación:

- Y es el valor de la etiqueta. Será 1 si la pareja de datos son de la misma clase o 0 si son de distinta clase.
- Y_{pred} es el valor de la etiqueta predicho por la red siamesa.

Sin embargo, la *contrastive loss* es una función de pérdida que se adapta mejor al problema que se quiere abarcar con las redes siamesas, ya que el objetivo de las mismas es diferenciar y no clasificar entre los dos elementos, como se observa en la ecuación 2.2.

$$Loss = Y * D^2 + (1 - Y) * \max(\alpha - D, 0)^2 \quad (2.2)$$

Descomponiendo la ecuación:

- Y vuelve a ser el valor de la etiqueta, al igual que en la Ecuación 2.1.
- D es la distancia euclídea entre la salida de cada red componente de la red siamesa.
- α es el margen, una distancia mínima que tienen que tener los puntos para ser considerados diferentes, penalizando muestras diferentes que estén muy cerca unas de otras. Por defecto está establecido a 1.

2.2.2. Tripletas

Las redes cuya entrada es una tripleta de parámetros o elementos surgieron en 2015 [6], a veces llamadas *Triplet Networks*. En ellas, a diferencia de tener una pareja de elementos donde se etiquetan si son iguales o diferentes, se tienen tres elementos. El primero de ellos se denomina ancla, que es el elemento el cual se va a comparar la similitud con los otros dos. Los otros dos son un elemento de la misma clase y un elemento de otra clase. Como se observa en la figura 2.2, así como en las redes cuya entrada es una pareja, las redes son idénticas, se comparten los pesos y se calcula la distancia entre muestras y se tiene una función de pérdida.

En este caso la función de pérdida no es ni la entropía binaria cruzada ni la *contrastive loss*, sino que para este problema se utiliza la *triplet loss* de la ecuación 2.3.

$$L(A, P, N) = \max(\|e(A) - e(P)\|^2 - \|e(A) - e(N)\|^2 + \alpha, 0) \quad (2.3)$$

Descomponiendo la ecuación:

- A, P y N son los parámetros de entrada del ancla, positivo y negativo, respectivamente.
- α es el margen entre las parejas positivas y negativas. Por defecto está establecido a 1.
- $e()$ son los vectores de características.

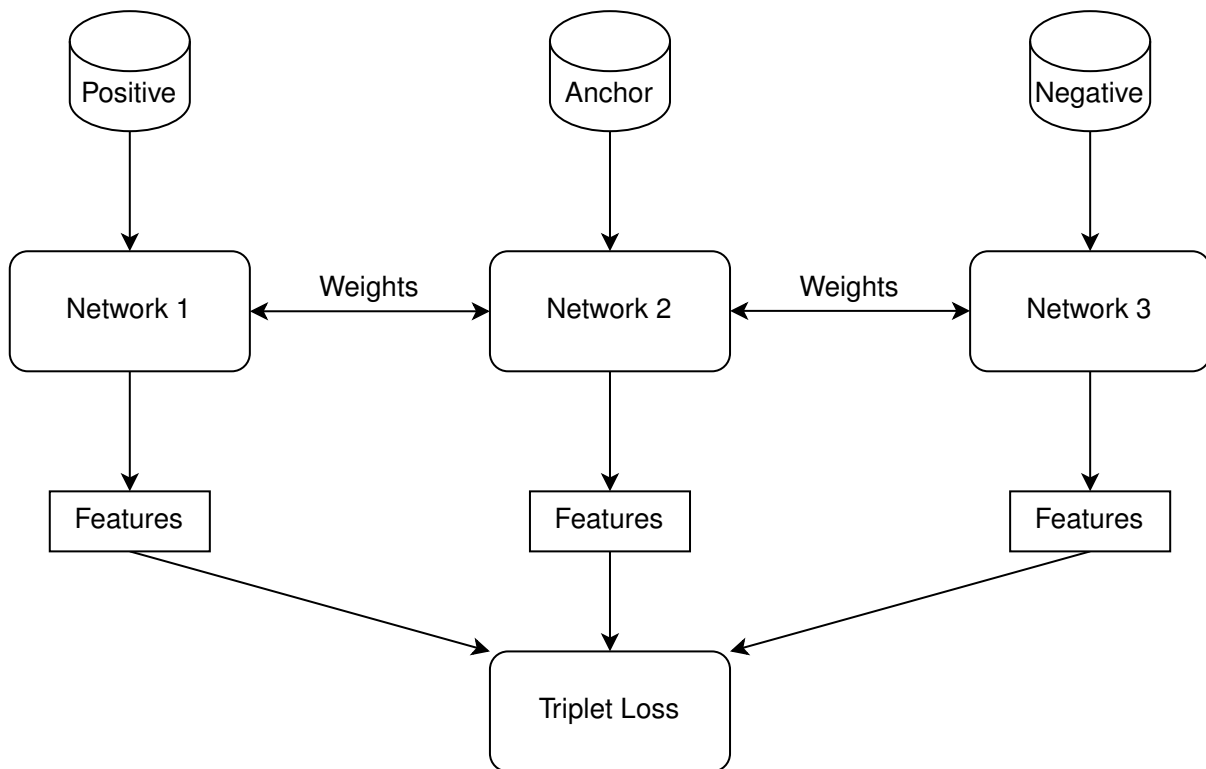


Figura 2.2: Arquitectura de Red siamesa de 3 entradas.

2.3. Integración de Sistemas de Recomendación con Redes Siamesas

Las SNN y los sistemas de recomendación llevan coexistiendo temporalmente como técnicas en la ingeniería informática desde principios de los 90, no obstante no se ha empezado a estudiar la integración de SNN junto a los sistemas de recomendación hasta más de dos décadas más tarde. En la bibliografía se encuentran los primeros artículos en 2018 [7–11] y en cada uno de ellos se encuentran diferentes aproximaciones para diferentes problemas.

2.3.1. Metodología

Se han recogido todos los artículos actuales disponibles para el estudio del estado del arte, realizando así el siguiente trabajo bibliográfico.

Primero se ha realizado una lluvia de ideas y pre-investigación, extrayendo los mejores conceptos y términos para las consultas en las bases de datos de artículos. Estas consultas se pueden encontrar en el apéndice A, son generalistas para poder extraer todos los artículos relacionados y luego más adelante filtrar entre válidos y no válidos. Se han utilizado las bases de datos de Web of Science <https://www.webofscience.com> y Scopus <https://www.scopus.com> extrayendo un total

de 55 artículos, de los cuales se han catalogado como válidos solo 24. Esta reducción de más de la mitad de los artículos es debida a la eliminación de artículos duplicados, apareciendo individualmente y como contenido de una conferencia, y que al realizar las consultas de una forma tan ambigua se obtenían resultados que no tenían que ver con sistemas de recomendación.

2.3.2. Tareas de Recomendación

Las SNN y los sistemas de recomendación son utilizados en diferentes dominios y con diferentes tipos de datos dependiendo del problema que se quiere abarcar. Cuando ambas se integran esto no cambia. En la bibliografía se encuentran diferentes variedades de dominios.

Se han tomado como referencia al hacer la clasificación que se presenta en la tabla 2.1 para catalogar los artículos, aquellos dominios de ejemplo incluidos por R. Burke y M. Ramezani en el libro de referencia de los sistemas de recomendación [12]; además debido a su alta representación, y su notable mención en la última edición del libro de referencia de sistemas de recomendación [13], se añade el dominio de la moda.

Dominio	Artículos	Texto	Audio	Imágenes	Vídeos
Búsqueda de Trabajo	[9, 14]	X			
Comercio Electrónico	[10, 15, 16]	X			
Moda	[7, 8, 17, 18]	X		X	
Música	[19, 20]		X		
Noticias	[11]	X			
Películas	[10, 21, 22]				
Turismo	[22, 23]				
Otros	[24–32]			X	X

Tabla 2.1: Catálogo de artículos según los diferentes dominios y tipos de datos.

Dependiendo del dominio se encuentran también diferentes tipos de datos que se utilizan para que aprendan los modelos y a la hora de realizar las recomendaciones, como también se puede ver en la tabla 2.1. Todos los artículos menos [14] emplean el uso de metadatos en los entrenamientos de los modelos. Destaca que en todos los artículos del dominio de la moda [7, 8, 17, 18] y en [26, 27] se emplean imágenes como parámetros de entrada de las redes siamesas. Por otro lado, se observa el uso de otros elementos multimedia, como audio, en los artículos del dominio de la música [19, 20] o el uso de vídeos en otros dominios [32]. Finalmente, se emplean también textos completos en diferentes dominios [7, 14] destacando el dominio de noticias [11]. No obstante, en algunos artículos del dominio de la moda [16, 17] se observa el uso de texto también en el entrenamiento del modelo, pero este es en la parte de recomendación y no de la red siamesa.

Así mismo, el uso de estos dominios y tipos de datos es común en sistemas de recomendación que usan técnicas de aprendizaje profundo, como se puede observar en un estudio de 2019 por S. Zhang, L. Yao, A. Sun, e Y. Tay [33]. En la segunda tabla de dicho artículo, se pueden observar el uso de tipos de datos: texto, audio, imágenes o videos; y el uso de los mismos en dominios: música, redes sociales o noticias, que son parecidos a los hallados en esta revisión. No obstante, también destaca el dominio de la moda, que ha recibido numerosas aportaciones en los últimos años, destacando así mismo en la nueva edición del libro de referencia de sistemas de recomendación [13].

2.3.3. Enfoque algorítmico

Las **SNN** en sí mismas son una herramienta que se utiliza para realizar la recomendación. Este uso puede variar entre distintos problemas, así como el diseño de la propia red. Estudiando la bibliografía se destacan cuatro categorías en las que catalogar el uso de redes siamesas: cuál es el uso de la red en el problema, número de los parámetros de entrada, función de pérdida y red *feedforward* utilizada en la red siamesa.

Uso de la red en el problema

En los sistemas de recomendación se ha observado que existen diferentes técnicas para implementar la funcionalidad de identificar qué artículos son útiles para el usuario. Así mismo, si se tratan las **SNN** como una caja negra que únicamente extrae la información de la similitud entre artículos, se pueden dividir su contribución en la recomendación en dos categorías: predicción y extracción de características.

En predicción se utiliza únicamente la salida de la propia red para establecer la similitud, y, por tanto, la importancia del artículo a recomendar. En estas aproximaciones se devuelven los N artículos con mayor puntuación para realizar la predicción.

En extracción de características se utiliza la salida de la propia red como un dato intermedio que se concatena a un modelo mayor que determina la importancia del artículo para generar las recomendaciones. Así mismo existen varias subdivisiones en esta categoría:

- *Feedforward*: En *feedforward* se utiliza la salida de la red junto con otros datos relevantes para hacer una recomendación mediante predicción de una red *feedforward*.
- Clusterización: En clusterización se utiliza la salida de la red como vector de características. Este es utilizado por alguna técnica de clusterización como K-means para hacer la recomendación.
- Aprendizaje de clasificación: En aprendizaje de clasificación se utiliza la salida de la red como vector de características. Este es utilizado por alguna técnica de aprendizaje de clasificación (*learning to rank*) como BPR para generar las recomendaciones.

En la tabla 2.2 se observa el catálogo de los artículos según los diferentes usos de la red.

Extracción de características	Predicción
Feedforward	[14, 18, 22, 24, 26, 31, 32]
Clusterización	[16, 20, 23, 27]
Aprendizaje de clasificación	[7, 17, 25]

Tabla 2.2: Catálogo de artículos según los diferentes usos de la red.

Parámetros de entrada

Se han observado dos modelos de redes siamesas basándose en el número de parámetros de entrada, parejas y tripletas. Esta división es la misma que se ha utilizado para catalogar por número de parámetros de entrada la integración de redes siamesas con sistemas de recomendación.

Es importante destacar qué se entiende por tripleta en las redes siamesas para recomendación, ya que en algunos artículos se establece que se usan erróneamente, siendo lo que se usa realmente parejas de datos etiquetadas. Una tripleta se compone de tres elementos ancla, positivo y negativo (A, P, N). El ancla es el artículo o usuario el cual se quiere aprender la distancia a los otros dos elementos de la tripleta. El positivo es un artículo de la misma clase al artículo ancla o un artículo que ha gustado al usuario ancla y el negativo es un artículo de distinta clase al artículo ancla o un artículo que no ha gustado al usuario ancla. En la figura 2.3 se observa cómo es el cambio de las distancias tras el entrenamiento de una tripleta, donde se reduce la distancia entre el ancla y el positivo y así mismo se separan el ancla y el negativo.

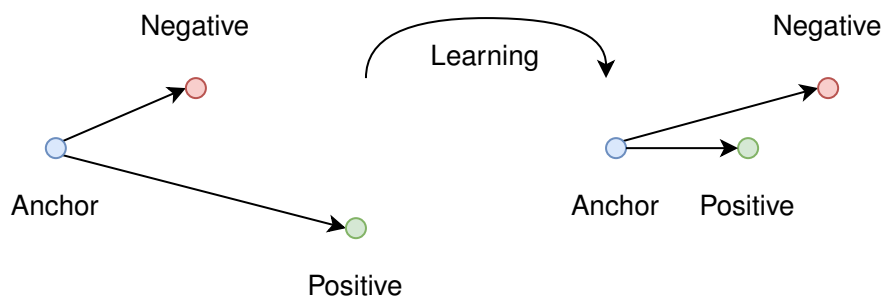


Figura 2.3: Cambio de las distancias en una tripleta tras el entrenamiento.

Es importante destacar también que los datos de la forma artículo 1, artículo 2, etiqueta (I_1, I_2, L) (que toma el valor 0 o 1 si los artículos pertenecen a la misma clase o no) no es una tripleta sino una pareja. Esta confusión se puede encontrar en *N2One: Identifying Coreference Object Among User Generated Content with Siamese Network* [22].

De los 24 artículos que se recogen en la tabla 2.1, 22 de ellos utilizan parejas y solo dos de ellos [19, 28] emplean tripletas, utilizando este último, además de tripletas, una novedosa estructura de parámetros de entrada de la forma: elementos ancla, un elemento positivo y n elementos negativos ($A, P, N_1, N_2, \dots, N_n$).

Función de pérdida

Los artículos también se pueden catalogar dependiendo de la función de pérdida que se utiliza para entrenar la red siamesa. Como se ha visto en la sección 2.2 existe una alta correlación entre la función de pérdida y el número de entradas a la red siamesa. Para todos los artículos que utilizan la entropía binaria cruzada [15, 20, 26] o la *contrastive loss* [8, 9, 14, 21, 22, 27, 32] usan parejas como parámetros de entrada, y para todos los artículos que utilizan la *triplet loss* [28] usan tripletas como parámetros de entrada. Sin embargo, no todos los artículos describen qué función de pérdida se ha utilizado, así como también utilizan otras funciones de pérdida, incluso funciones de pérdida personalizadas [23, 25, 29].

Entre las otras funciones de pérdida se encuentra *Softmax-cross-entropy* [10], *Point-wise loss* [11], *Multiple Negative Ranking Loss* [30], *Max-margin hinge loss* y *Categorical cross-entropy loss* [19].

Red feedforward

Por último se pueden catalogar los artículos dependiendo de la red *feedforward* que se usa en la red siamesa, como se observa en la tabla 2.3

Red <i>feedforward</i>	Artículos
Perceptrón multicapa	[21, 24, 25, 28, 29, 31]
Red convolucional	[7–9, 14, 16–20, 26, 27, 32]
Red recurrente	[10, 11, 14–16, 22, 23]
Transformer	[30]

Tabla 2.3: Catálogo de artículos según las diferentes familias de red *feedforward*.

Existen diferentes aproximaciones más concretas dentro de cada categoría de red *feedforward*. Dentro de perceptrón multicapa, sobresale el uso de *multi-armed bandits* [31] como técnica de aprendizaje por refuerzo. En las redes convolucionales destaca el uso de diferentes modelos: VGG-16 [18, 27], Inception V3 [7, 32], AlexNet [17] y C3D [32] en el uso de vídeos. En las redes recurrentes destacan diferentes arquitecturas: GRU [22], LSTM [10, 11, 14, 16, 23] y Bi-LSTM [14, 15]. Por último, el modelo de transformer que se utiliza en el único artículo en la categoría es SciBERT.

2.3.4. Ajustes de evaluación

Tanto para medir el rendimiento de redes neuronales artificiales como de sistemas de recomendación se utilizan diferentes técnicas de evaluación. Con estas se pueden comparar diferentes aproximaciones de modelos para el mismo problema, detectando cuál es el mejor algoritmo para resolverlo o para detectar cómo de bien realiza su función el algoritmo.

Cada uno de estos artículos tienen objetivos distintos (por ejemplo, lidiar con el efecto *long tail* [21] o de arranque en frío [20]) y, por lo tanto, miden cosas distintas, no obstante, se encuentran similitudes en algunas métricas y técnicas que se utilizan. En el manual de sistemas de recomendación [1] destacan dos tipos de técnicas de evaluación para los sistemas de recomendación: *offline* y *online*.

Las técnicas *offline* se realizan con datos ya recogidos intentando simular el comportamiento de los usuarios, mientras que las técnicas *online* comparan la interacción de varios sistemas de recomendación con usuarios reales, observando cómo influyen en los mismos. Las técnicas *online* son más costosas que las técnicas *offline* y no son reproducibles, no permitiendo comparar otros sistemas de recomendación con los evaluados si no se han incluido en el experimento. Es por esto, por lo que en la bibliografía de los 24 artículos solo se encuentran dos que utilicen técnicas de evaluación *online* [15, 20].

Sin embargo, todos los artículos utilizan técnicas de evaluación *offline*. No existe ninguna métrica concreta que sea mejor que el resto, depende de lo que se quiera medir [34]. En la tabla 2.4 se observa el catálogo total de artículos junto a las diferentes métricas de evaluación que se utilizan en cada uno. Se puede observar que las técnicas que más se emplean son Recall@K, Precision@K, Accuracy, Area Under the ROC Curve (AUC), F1, Normalized Discounted Cumulative Gain (NDCG), Mean Reciprocal Rank (MRR) y Hit Ratio (HR). En concreto, todas estas técnicas tienen en común que son métricas de evaluación de ranking, donde evalúan la calidad de la lista devuelta por el sistema de recomendación, en lugar del acierto en la predicción.

Artículo	Métrica de evaluación
Año 2018	
[7]	AUC
[8]	Mean Recall@K
[9]	Accuracy, Precision, Recall, F1
[10]	Recall@K, MRR
[11]	HR@K, NDCG@K
Año 2019	
[17]	AUC
[18]	Lift@K
[21]	Precision, Recall, F1, BPREF, LTC, WLTC, TTC
[29]	AUC, ERR, RD, FPR, TPR, FNR
[32]	Recall, Betrayal Rate
Año 2020	
[14]	Accuracy
[19]	Precision, AUC
[22]	Precision
[23]	Accuracy@K, macro-F1
[25]	HR, NDCG
[26]	Precision, Recall, F1, ROC
[27]	Accuracy
Año 2021	
[15]	Precision, Recall, A/B Test
[16]	Accuracy, AUC
[20]	Accuracy, Satisfacción de Usuario
Año 2022	
[24]	AUC, NDCG, MMR, PR-AUC
[28]	Precision, Recall, HR and Average Reciprocal HR
[30]	Precision, Recall, F1, MRR, MAP
[31]	Accuracy@K, Precision@K, Recall@K, F1@K, ROC

Tabla 2.4: Catálogo de artículos junto a sus métricas de evaluación.

IMPLEMENTACIÓN DE LOS MODELOS

En este cuarto capítulo se define el proceso de desarrollo de los experimentos que se quieren llevar a cabo. Se desarrollan 3 experimentos para observar la reproducibilidad de la integración de SNN y sistemas de recomendación. De los 3 experimentos, los primeros dos se intentan ajustar a las técnicas de recomendación basadas en contenido y el último a las técnicas basadas en filtrado colaborativo. No obstante, en los experimentos del estado del arte se observa la complejidad de ceñirse o catalogarse puramente en estas estrategias, no diciendo nada o implementando una estrategia híbrida.

Así mismo, tras estudiar diferentes tecnologías, se ha decidido utilizar un entorno en Python, basándose en las librerías TensorFlow [35] y Keras [36] para los experimentos. Además, es donde se ha encontrado el código mejor documentado, haciendo más sencilla la reutilización por parte de terceros. A su vez, son las librerías que más han sido utilizadas en los artículos del estado del arte donde tenemos acceso al código, como observamos en la tabla 3.1.

Artículo	Año	Librerías	Link
[10]	2018	TensorFlow	https://github.com/PreferredAI/cbs
[11]	2018	Keras	https://github.com/dhruvkhattar/RARE
[16]	2021	TensorFlow, Keras	https://github.com/marinaangelovska/complementary_products_suggestions
[20]	2021	TensorFlow, Keras	https://github.com/michaelpulis/SnnForCbColdStartMusicRecommendation
[30]	2022	PyTorch	https://github.com/malteos/aspect-document-embeddings

Tabla 3.1: Catálogo de artículos con código.

La arquitectura del problema a abarcar es igual en todos los experimentos que se van a crear, siguiendo la estructura que se observa en la figura 3.1. Este tipo de estructura para estos experimentos se cataloga para el apartado “uso de la red en el problema” como predicción.

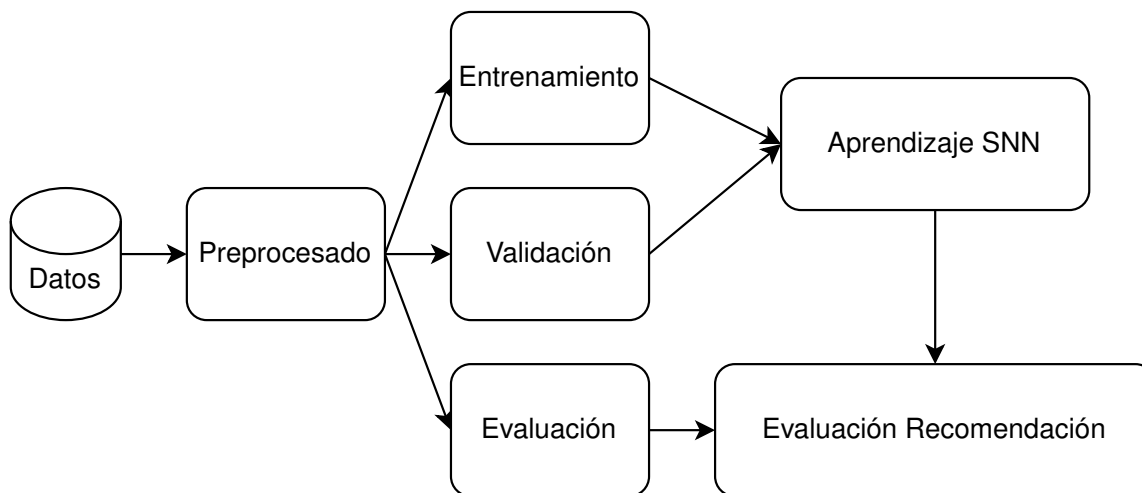


Figura 3.1: Arquitectura de un problema de recomendación con redes siamesas.

Primero se tienen que buscar unos datos acordes al problema que queremos abordar. Estos datos necesitan tener unas características concretas dependiendo de si la recomendación es basada en contenido o de filtrado colaborativo, necesitando información o no de los artículos, etiquetado de los mismos, usuarios, puntuaciones, etc.

Se continúa con el preprocesamiento de los datos. Por eficiencia, se hace un estudio de los datos viendo la distribución de los mismos, las diferentes etiquetas que tienen. En este proceso se pueden eliminar diferentes valores atípicos que puedan tener consecuencias negativas en el aprendizaje de los modelos. Aquí es importante tener en cuenta qué se establece como similitud y diferencia entre artículos, así como establecer si un artículo es positivo o negativo para cada usuario. Por último, en el preprocesado de los datos y dependiendo del problema, se crean parejas o tripletas con los datos obtenidos.

Después, se particionan los datos en entrenamiento, validación y evaluación. Cada uno de estos conjuntos de datos se utiliza en las siguientes etapas del aprendizaje del sistema de recomendación, utilizándose los conjuntos de entrenamiento y validación en el aprendizaje de la red siamesa y los datos de evaluación junto a esa red siamesa aprendida, en la evaluación del sistema de recomendación.

Posteriormente, se entrena o ajusta, si se está utilizando un modelo preentrenado de red, la red siamesa con los datos de entrenamiento. Durante el entrenamiento se utilizan los datos de validación para comprobar que la red no está sobreaprendiendo, generalizando correctamente. Además, estos datos de validación permiten establecer una condición de parada temprana del entrenamiento si este no ha aprendido nada tras un número de iteraciones.

Finalmente, se evalúa el modelo con los datos de evaluación. Para la evaluación se decide utilizar métricas de Recall@K y Precision@K, denominadas también métricas de ranking, evaluando así las K primeras recomendaciones que ofrece el modelo, en vez de la habilidad del modelo de crear las recomendaciones, pues en una recomendación real solamente se muestran al usuario de 1 a 10 artículos normalmente.

3.1. Experimentos Basados en Contenido

Se desarrollan dos experimentos de recomendación basados en contenido con redes siamesas con parejas como parámetros de entrada. Observando la bibliografía se aprecia que este tipo de experimentos no suele utilizar datos de usuarios, sino los datos de los artículos, comparando en sí la similitud entre ellos y devolviendo los artículos más similares al artículo evaluado. Para ello, se va a realizar dos experimentos, uno con imágenes y otro con audios, comprobando así la recomendación con ambos tipos de datos.

Primero, los datos tienen que estar etiquetados, es decir, si tenemos un artículo del dominio de la moda, qué tipo de artículo es: camiseta, calzado, pantalón, gorro, etc. El nivel y especificidad del etiquetado debe ser acorde a qué se quiere comparar y a qué se quiere denominar como similar o diferente por el autor del problema.

Una vez con los datos etiquetados y tras su preprocesado, estos se emparejan y si ambos comparten etiqueta, se etiqueta la propia pareja con un 1 marcando que son una pareja de artículos similares y si no, se etiqueta la pareja con un 0 marcando que son una pareja de artículos diferentes.

Así mismo, los datos pueden tener varias etiquetas que pueden resultar útiles a la hora de decir si una pareja de artículos son similares o no. Por ejemplo, si queremos decir si dos alimentos son similares por el tipo de sabor primario que nos ofrecen, podemos encontrar alimentos puramente de un tipo de sabor primario (la sal es salada); y alimentos que pueden etiquetarse con varios sabores primarios (la naranja es dulce y ácida). Dependiendo del problema que se quiere abarcar, se pueden tomar varias aproximaciones: establecer que son iguales si todas y cada una de las etiquetas coinciden o establecer que son iguales si al menos una o x número de etiquetas coinciden.

En los experimentos planteados se quiere comparar cómo afectan diferentes modelos de redes *feedforward* en la red siamesa en la recomendación, así como observar la diferencia entre el uso de las distintas funciones de pérdida. Como algoritmo a utilizar se van a comparar en concreto varias **SNN** con dos imágenes como parámetros de entrada prediciendo las recomendaciones. Para ello se comparan cómo funciona la entropía binaria cruzada y *contrastive loss* teniendo como red *feedforward* diferentes arquitecturas de redes convolucionales preentrenadas en ImageNet, VGG-19 e Inception-ResNet v2 y una red convolucional propuesta por A. Rosenbrock [37] para el conjunto de datos MNIST [38] como la de la figura 3.2. Es decir, se tendría una arquitectura como la de la figura 2.1 teniendo

como *Input 1* e *Input 2* cada imagen de entrada perteneciente a la pareja, como *Network 1* y *Network 2* las redes *feedforward* mencionadas anteriormente y como *Loss Function* la entropía binaria cruzada y *contrastive loss*.

Así mismo, durante el entrenamiento de los modelos, los datos de entrenamiento se dividen entre entrenamiento y validación. Sirviendo los datos de validación para que el modelo no sobreentrene generalizando correctamente, teniendo en cuenta la evolución de la pérdida en validación, y para realizar una parada temprana si no ha aprendido nada en 10 épocas.

Teniendo en cuenta el enfoque algorítmico en 2.3.3, estos experimentos se catalogarían como: uso de la red, predicción; parámetros de entrada, pareja; función de pérdida, entropía binaria cruzada y *contrastive loss* y por último, red *feedforward*, red convolucional.

3.1.1. Dominio de moda

Para el primer experimento, en la última edición del libro de referencia de sistemas de recomendación [13] se observa que se expone el uso de redes siamesas en el dominio de la moda. Así mismo, este primer experimento consiste en la recomendación basada en contenido con imágenes como datos de entrada. Para ello, lo primero que se ha hecho ha sido adaptar dos implementaciones que se han encontrado. La primera implementación surge en un blog de A. Rosenbrock [37], donde explica paso a paso y con detenimiento el uso de **SNN**, cómo entrenarlas, qué métricas usar y su implementación en Tensorflow. La segunda es de un Kaggle de V. Luhaniwal [39], donde realiza un sistema de recomendación con imágenes, pero usando redes neuronales artificiales convolucionales, no **SNN**. Esta última aproximación también propone un conjunto de datos de moda compatible con los requisitos del experimento [40], con imágenes de ropa para el torso, ropa para las piernas y calzado; clasificado también entre hombre y mujeres.

3.1.2. Dominio de música

Para el segundo experimento de recomendación basada en contenido con audios como datos de entrada, se toma como referencia un artículo del estado del arte [20]. En este artículo, que incluye parte de código, se observa un sistema de recomendación en el dominio de la música con datos de audios que son transformados en imágenes mediante la captura de su espectrograma de Mel y, a continuación, pasados a través de un sistema de recomendación con **SNN** integradas para realizar la recomendación. A su vez, este artículo propone el conjunto de datos de audios etiquetados compatible con los requisitos del experimento, FMA [41, 42]. En este la mayor diferencia que se puede encontrar con el primer experimento es en el preprocesado de los datos, tratando los audios tras un preprocesado como imágenes y luego ya utilizando un sistema de recomendación con redes siamesas de imágenes. En este caso se utiliza el mismo sistema que en el primer experimento, pudiendo observar que nuestro

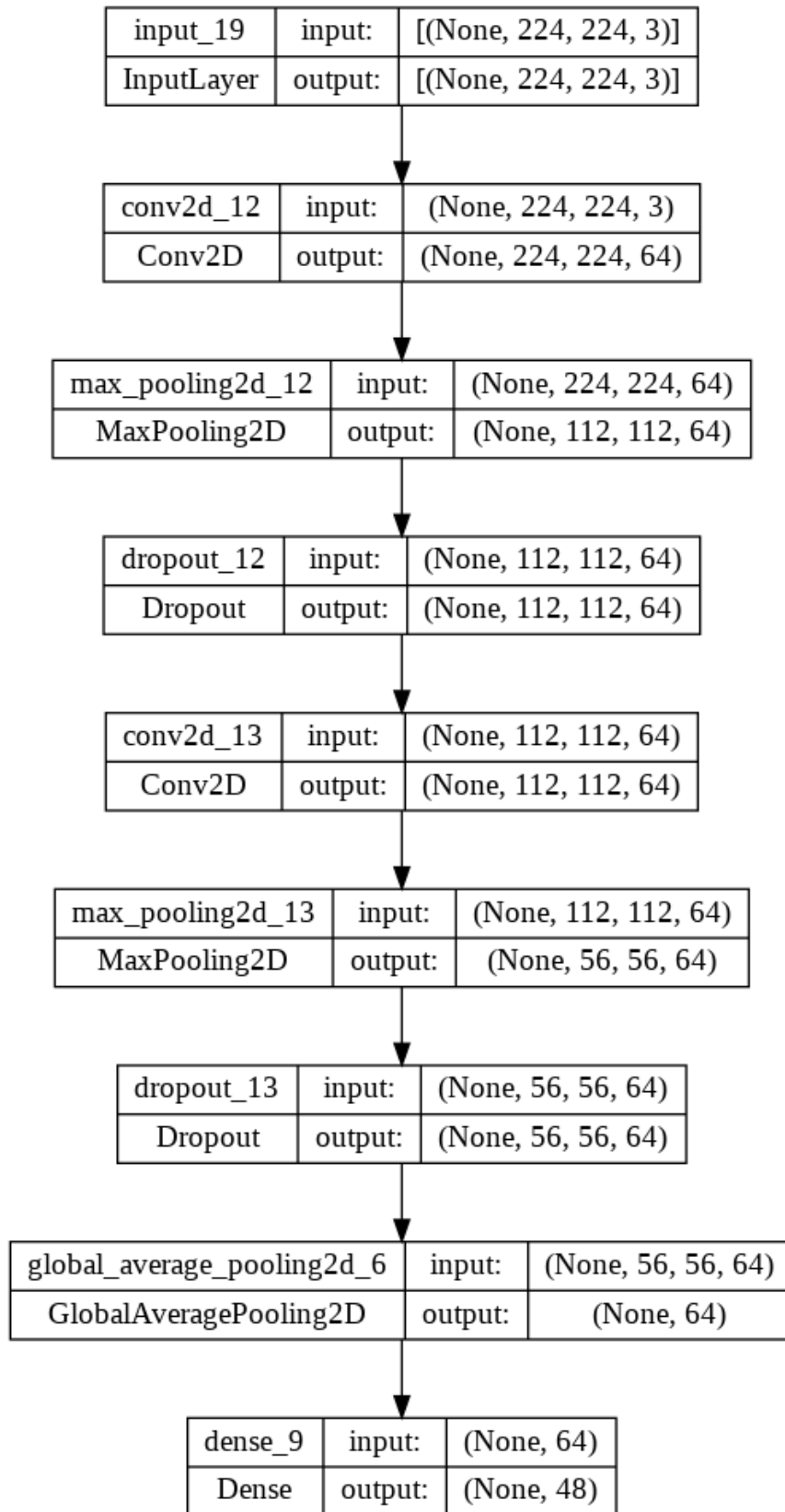


Figura 3.2: Arquitectura de la red convolucional propuesta en el conjunto de datos MNIST.

sistema de recomendación es reutilizable por diferentes tipos de datos y dominios, siendo lo importante el preprocesado de los datos, como se establece que es similar entre los diferentes artículos y la creación de las parejas.

3.2. Experimentos Filtrado Colaborativo

Se desarrolla un experimento de recomendación basada en contenido con redes siamesas con tripletas como parámetros de entrada. Para ello, primero los datos tiene que contener un historial de usuarios como ya se ha visto en 2.1.2, es decir, un conjunto de datos de usuarios con respuestas de los usuarios a los artículos, pudiendo ser estas escalares, binarias y unarias.

Una vez con los datos y tras su preprocesado, se crea la tripleta (Usuario, Artículo positivo, Artículo negativo). El autor del experimento debe especificar qué se entiende como un artículo positivo y negativo para el usuario. Como se ha establecido, la respuesta que tienen los usuarios a los artículos pueden variar, siendo esta escalar, binaria y unaria. Si la respuesta es escalar, se debe establecer a partir de qué puntuación se establece si el artículo es positivo o negativo. En el caso de la respuesta binaria es más sencillo, pues estas suelen establecer si le ha gustado o no al usuario, pudiendo utilizar esto para catalogar los artículos como negativos o positivos. Por último, con las respuestas unarias, se puede establecer que todos los artículos marcados son positivos, pero establecer que el resto de artículos son negativos es una hipótesis demasiado fuerte, por lo que se suele hacer un *sampling* para elegir al azar un número limitado de artículos negativos.

Como algoritmo a utilizar se va a comparar varias **SNN** con tripletas como datos de entrada prediciendo las recomendaciones frente a un modelo de sistema de recomendación de filtrado colaborativo de vecindarios basado en usuarios. Para ello se usa la *triplet loss* como función de pérdida, se tiene como red *feedforward* una arquitectura de perceptrón multicapa. Es decir, se tendría una arquitectura como la de la figura 2.2 teniendo como *Anchor* el usuario que recibe la recomendación, como *Positive* un artículo valorado positivamente por el usuario y como *Negative* un artículo valorado negativamente por el usuario. Además, como *Network 1*, *Network 2* y *Network 3* se tiene el perceptrón multicapa.

Teniendo en cuenta el enfoque algorítmico en 2.3.3, este experimento se catalogaría como: uso de red, predicción; parámetros de entrada, tripleta; función de pérdida, *triplet loss* y por último, red *feedforward*, perceptrón multicapa.

En el experimento se quiere comparar cómo afecta el número de elementos que recomienda nuestro sistema con respecto a Precision y Recall, así como comparar nuestro sistema con un modelo de sistema de recomendación clásico de vecindarios basado en usuario.

Este sistema de recomendación clásico de vecindarios basado en usuario primero calcula la similitud entre el usuario al cual se quiere recomendar y el resto de usuarios según los datos de en-

trenamiento de los artículos que han evaluado entre ambos. En este caso se ha utilizado la similitud coseno para calcular la similitud entre usuarios. Incorpora una mejora de mínimo solapamiento de artículos, frente a otros sistemas de recomendación basados en usuarios. Esta mejora permite dar más importancia a los usuarios que hayan valorado un mayor número de artículos en común, no teniendo únicamente en cuenta la puntuación que se le ha dado a esos artículos.

Después, se crean los vecindarios con los x usuarios más similares al usuario que recibe la recomendación. Así mismo se tiene que estudiar cuál es el número óptimo de usuarios en el vecindario para realizar las recomendaciones en el problema. Por último, se pondera la similitud de los usuarios en el vecindario con la puntuación normalizada que han dado al artículo para conseguir la puntuación del usuario objetivo para ese artículo, como es habitual en la literatura [43].

EXPERIMENTOS Y RESULTADOS

En este cuarto capítulo se explican detenidamente los experimentos realizados, qué conjuntos de datos se han utilizado y qué resultados se han obtenido

Se han desarrollado 3 experimentos para observar la reproducibilidad de la integración de SNN y sistemas de recomendación. Los experimentos se han realizado en cuadernos de Jupyter mediante Google Colab. Este entorno permite la ejecución de los cuadernos proporcionando acceso a un ordenador con características similares a las que observamos en la tabla 4.1.

Recursos	Características
CPU	Intel(R) Xeon(R)
Frecuencia CPU	2.20GHz
Nº de núcleos CPU	2
Tamaño de disco	80GB
GPU	Nvidia Tesla T4
Memoria GPU	16GB
RAM	12.6 GB

Tabla 4.1: Especificaciones Hardware del entorno utilizado en Google Colab.

Los cuadernos de Jupyter creados para los experimentos pueden ser utilizados para el desarrollo de otros sistemas de recomendación con redes siamesas para otros datos de entrada. Para acceder a ellos, estos han sido subidos a un repositorio de GitHub <https://github.com/masternico97/Siamese-Recommender-Systems>.

4.1. Experimento con imágenes

Se realiza el primer experimento teniendo en cuenta la implementación de los modelos para experimentos basados en las descripciones del apartado 3.1.2, así como los experimentos realizados por A. Rosenbrock [37] y V. Luhaniwal [39].

4.1.1. Configuración

En este experimento se utiliza el conjunto de datos del Kaggle de V. Luhaniwal [39], donde propone un conjunto de datos de moda [40], con imágenes de ropa para el torso, ropa para las piernas y calzado; clasificado también entre hombres y mujeres. Se escogen únicamente los datos catalogados como calzado para hombre, en total 811 artículos. Estos a su vez están divididos en 3 categorías: zapatos, chanclas y sandalias; y en 6 subcategorías: zapato casual, zapato formal, zapato de deporte, chanclas, sandalia y sandalia deportiva; siendo estas últimas las que se utilizan para etiquetar los datos manualmente como perteneciente a la misma clase o no. Así mismo también tenemos la información del color del calzado, pero no se tiene en cuenta como etiqueta.

Se extrae la información de los datos que se va a necesitar para comparar la similitud, en nuestro caso los píxeles de las imágenes de la dirección URL que aporta el conjunto de datos. A continuación, se dividen los datos en un 70 % datos de entrenamiento y un 30 % datos de evaluación. Así mismo, se establece una semilla de esta partición aleatoria para poder reproducir el experimento más adelante. Después, se hacen las parejas teniendo en cuenta el etiquetado de la subcategoría tanto de los datos de entrenamiento como los de evaluación.

4.1.2. Resultados

Para la evaluación de este experimento, se ha optado por la aproximación que podemos encontrar en otro artículo centrado únicamente en sistemas de recomendación de calzado basado en imágenes [44]. Se extraen aleatoriamente una muestra estadísticamente significativa de las imágenes del conjunto de datos y se examinan las K mejores recomendaciones para cada imagen. A diferencia de en artículo anteriormente citado, en vez de dar puntuaciones a dichas recomendaciones y hacer la media de todas las puntuaciones para calcular Accuracy, se calcula Recall@10 y Precision@10. Se destaca que estos conjuntos aleatorios también tienen una semilla para que sea comparable el Recall@K y Precision@K entre los diferentes modelos.

	Entropía Binaria Cruzada	Contrastive loss
MNIST CNN	0.0114	0.0247
VGG-19	<u>0.0551</u>	0.0200
Inception-ResNet v2	0.0114	0.0230

Tabla 4.2: Resultados de Recall@10 para el experimento 1.

Teóricamente y según lo leído en el estado del arte, en las tablas en las tablas 4.2 y 4.3 para el mismo modelo debería dar mejores resultados la columna de *Contrastive loss* frente a la columna de *Entropía Binaria Cruzada*, debido a que la *contrastive loss* es una función de pérdida que se adapta mejor al problema que se quiere abarcar con las redes siamesas, ya que el objetivo de las mismas

	Entropía Binaria Cruzada	Contrastive loss
MNIST CNN	0.3025	0.3850
VGG-19	<u>0.8100</u>	0.2925
Inception-ResNet v2	0.1725	0.4050

Tabla 4.3: Resultados de Precision@10 para el experimento 1.

es diferenciar y no clasificar entre los dos elementos. Por el contrario, para el modelo VGG-19 se encuentra que tanto Recall@10 como Precision@10 es mayor en la columna de *Entropía Binaria Cruzada* indicando que presenta mejores resultados que cuando se utiliza la *contrastive loss*. Así mismo, también se puede observar que para los otros modelos esta propiedad sí que se cumple, por lo que no se puede determinar qué función de pérdida es mejor en los sistemas de recomendación con la integración de redes siamesas.

Además, el modelo VGG-19 utilizando como función de pérdida la entropía binaria cruzada es el que arroja mejores resultados en este problema, teniendo valores de Recall@10 de 0.0551 y Precision@10 de 0.81.

4.2. Experimento con audios

Se realiza el segundo experimento teniendo en cuenta la implementación de los modelos para experimentos basados en contenido del apartado 3.1.2, así como el código del artículo [20] y lo anteriormente desarrollado para el primer experimento.

4.2.1. Configuración

Se utiliza el conjunto de datos que propone [20], FMA [41, 42], compuesto por datos, metadatos y extractos de 30 segundos de audios de canciones de 16 géneros diferentes. Estos 16 géneros son los que se utilizan como etiquetas de los datos para crear las parejas, estableciendo si ambas canciones pertenecen a la misma clase o no.

En este experimento primero se transforman los audios en imágenes extrayendo su espectrograma de Mel. A continuación, mediante el nombre de las imágenes, extraemos su información del archivo de metadatos para conocer a qué género pertenecen y saber qué etiqueta tienen. A continuación, se dividen los datos en un 70% datos de entrenamiento y un 30% datos de evaluación. Así mismo, se establece una semilla de esta partición aleatoria para poder reproducir el experimento más adelante. Después, se hacen las parejas teniendo en cuenta el etiquetado del género tanto de los datos de entrenamiento como los de evaluación. Como las canciones pueden pertenecer a varios géneros musicales, a la hora de hacer parejas para saber si dos canciones son del mismo género o no, en este caso se establece que son del mismo género si al menos un género de alguna de las canciones coincide.

4.2.2. Resultados

Se usan las mismas métricas de evaluación Recall@10 (ver tabla 4.4) y Precision@10 (ver tabla 4.5) sobre un conjunto de datos aleatorio estadísticamente significativo, como en el primer experimento. Se destaca que estos conjuntos aleatorios, también tienen una semilla para que sea comparable el Recall@K y Precision@K entre los diferentes modelos.

	Entropía Binaria Cruzada	Contrastive loss
MNIST CNN	<u>0.0764</u>	0.0552
VGG-19	0.0130	0.0244
Inception-ResNet v2	0.0268	0.0269

Tabla 4.4: Resultados de Recall@10 para el experimento 2.

	Entropía Binaria Cruzada	Contrastive loss
MNIST CNN	0.2607	<u>0.3000</u>
VGG-19	0.1107	0.1893
Inception-ResNet v2	0.2071	0.1429

Tabla 4.5: Resultados de Precision@10 para el experimento 2.

Así como en el experimento anterior, hemos observado que para el modelo VGG-19 se encuentra que tanto Recall@10 como Precision@10 es mayor en la columna de *Entropía Binaria Cruzada* indicando que presenta mejores resultados que cuando se utiliza la *contrastive loss* en este experimento este comportamiento es más irregular aún. En Precision@10, para MNIST CNN parece comportarse mejor la entropía binaria cruzada y en VGG-19 e Inception-ResNet v2 parece arrojar mejores resultados la *contrastive loss*. Mientras que para Recall@10, MNIST CNN y VGG-19 parece comportarse mejor *contrastive loss* y en Inception-ResNet v2 parece arrojar mejores resultados la entropía binaria cruzada

A su vez, solo se puede determinar con exactitud qué modelo es el que tiene mejores resultados, siendo este MNIST CNN, pero no qué función de pérdida se adapta mejor a este modelo, pues depende de si queremos mejorar Precision o Recall del mismo. Es interesante observar que el modelo MNIST CNN proporciona mejores resultados que los otros dos modelos, siendo estos más complejos. Esto puede ser debido a que esos dos modelos están preentrenados con Imagenet, un conjunto de imágenes etiquetadas de objetos, pero no entrenada para el procesamiento de espectrogramas.

4.3. Experimento en el dominio de películas

Se realiza el tercer experimento teniendo en cuenta la implementación de los modelos para experimentos basados en las descripciones del apartado 3.2. Además, se adapta una implementación de un

Kaggle de S. Golovkine [45] como implementación de un sistema de recomendación con tripletas en un curso de Máster en Ciencia de Datos del Institut Polytechnique de París. En dicha implementación, a diferencia de este experimento en el que se quiere utilizar el dominio de películas, se propuso un conjunto de datos de recomendación de series de animación japonesas [46].

4.3.1. Configuración

El tercer experimento consiste en recomendación utilizando la técnica de filtrado de colaborativo en el dominio de películas. Para ello, se va a utilizar MovieLens Small [47], un conjunto de datos de recomendación de películas con 100836 *ratings*, 9724 películas y 610 usuarios.

Para el experimento se extraen primero las películas con al menos 50 *ratings*, reduciendo el conjunto de datos a 40660 *ratings*. Así mismo, se extraen los usuarios que han interactuado con esas películas y el número de películas resultantes, siendo estos 606 usuarios y 436 películas. Una vez preprocesado los datos se dividen en datos de entrenamiento y evaluación, siendo la ratio 80 % datos de entrenamiento y 20 % datos de evaluación. Así mismo, se establece una semilla de esta partición aleatoria para poder reproducir el experimento más adelante. Después, se hacen las tripletas (usuario, artículo positivo, artículo negativo) teniendo en cuenta la puntuación que el usuario ha otorgado a la película, estableciendo que una película es positiva si tiene una puntuación mayor a 3,5.

4.3.2. Resultados

Para la evaluación de este experimento, en vez de utilizar la **AUC**, como se indica en el experimento en el que se basa, se utiliza **Recall@K** y **Precision@K**. Este cambio lo realizamos para una mejor comparación con el resto de apartados de este trabajo y a que, como se indica al final del propio artículo, en la parte de “posibles extensiones”, se establece que: “en el notebook se ha evaluado la calidad del ranking de recomendación utilizando la métrica **AUC**. Esta métrica refleja la habilidad del modelo de crear un ranking correctamente para cualquier par de artículos. Sin embargo, en la práctica al solo mostrar una serie de artículos a los usuarios, normalmente de 1 a 10, es más informativo utilizar métricas que tengan en cuenta los primeros artículos de la recomendación, como la **Precision@K**.”

Así mismo, se quiere estudiar cómo afecta el número de elementos que recomienda nuestro sistema, la K , en las métricas **Recall@K** y **Precision@K** como se observa en la tabla 4.6 y en la gráfica 4.1. Para ello, ya que en este experimento, a diferencia de los dos anteriores, requiere menos costo computacional y en vez de utilizar una muestra representativa del total de los usuarios, se utilizan todos ellos en el cálculo de las métricas de evaluación.

Se puede observar que a mayor el número de artículos devueltos por el modelo, se tiene un mayor **Recall** y una menor la **Precision** y que cuanto menor sea el número de artículos devueltos por el modelo, menor es el **Recall** y mayor la **Precision**. Esto es importante porque dependiendo de para qué

K	Recall@K	Precision@K
1	0.0307	0.2065
2	0.0571	<u>0.2083</u>
3	0.0849	<u>0.2083</u>
4	0.1097	0.1993
5	0.1265	0.1860
6	0.1414	0.1765
7	0.1631	0.1713
8	0.1829	0.1685
9	0.2002	0.1612
10	<u>0.2190</u>	0.1587

Tabla 4.6: Resultados de Recall@K y Precision@K en el experimento 3 para el sistema de recomendación con la integración de la red siamesa, diferentes valores de K.

se quiera usar el modelo, se puede escoger desarrollar un modelo con mayor Recall, mayor Precision o que haga un balance entre ambas métricas.

Por último, el sistema de recomendación basado en redes siamesas se compara con un sistema de recomendación de vecindarios basado en usuarios. Primero, se estudia cuál es el mejor tamaño de vecindario en nuestro sistema de recomendación, teniendo en cuenta las métricas de Recall@K y Precision@K para un valor de $K = 10$, como se observa en la tabla 4.7.

Tamaño del Vecindario	Recall@10	Precision@10
1	0.1024	0.0640
3	0.1203	<u>0.0754</u>
5	0.1309	0.0749
7	<u>0.1348</u>	0.0715
11	0.1196	0.0668
19	0.1012	0.0690

Tabla 4.7: Resultados de Recall@10 y Precision@10 con diferentes tamaños de vecindario para el sistema de recomendación basado en usuarios.

Se observan que los tres mejores valores de Precision@10 y Recall@10 se obtienen para tamaños de vecindario 3, 5 y 7. Se elige por tener valores mejores en media, el tamaño de vecindario 5 como el óptimo. Una vez observado cuál es el tamaño de vecindario óptimo, se calculan las métricas de Recall@K y Precision@K para valores de K entre 1 y 10, como en el sistema de recomendación con redes siamesas (ver tabla 4.8). Se puede apreciar que se obtienen mejores resultados en el sistema de recomendación que tiene integrada la red siamesa, ya que su peor valor de Precision es 0.15, mientras que el mejor valor de recomendador basado en vecinos es 0.08 (para distintos valores de K,

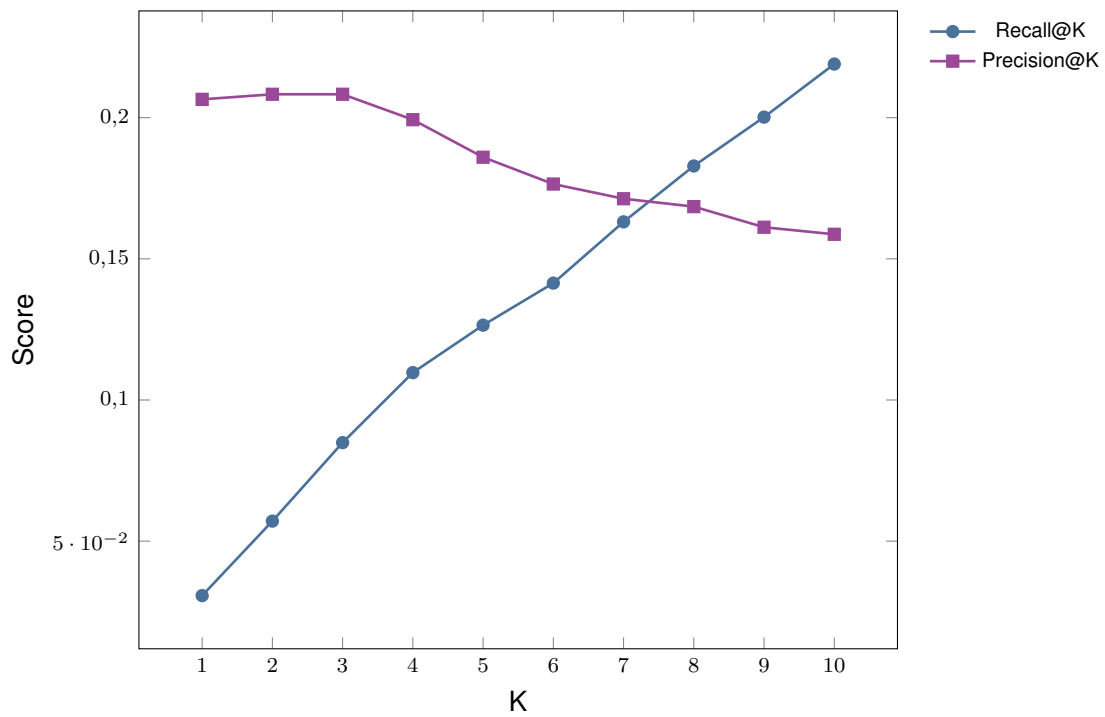


Figura 4.1: Gráfica Recall@K frente Precision@K experimento 3

K	Recall@K	Precision@K
1	0.0139	0.0735
2	0.0273	0.0726
3	0.0483	0.0783
4	0.0645	<u>0.0802</u>
5	0.0762	0.0785
6	0.0893	0.0797
7	0.1033	0.0776
8	0.1128	0.0775
9	0.1234	0.0767
10	<u>0.1309</u>	0.0749

Tabla 4.8: Resultados de Recall@K y Precision@K en el experimento 3 para el sistema de recomendación de vecindarios basado en usuario, con un tamaño de vecindario 5.

de hecho). Así mismo, se observan ventajas en los tiempos de ejecución del entrenamiento, tanto del sistema con la red siamesa frente al sistema basado en usuarios, siendo estos de 2 minutos en el caso del sistema de recomendación con la red siamesa integrada, y de más de 20 minutos para el sistema de recomendación de vecindarios basado en usuarios.

CONCLUSIONES Y TRABAJO FUTURO

Por último, se exponen las conclusiones de este proyecto, así como las posibles ampliaciones que se puede realizar teniendo como base este proyecto, tanto sobre los experimentos realizados como con la investigación del mismo.

5.1. Conclusiones

El uso de las redes siamesas junto a los sistemas de recomendación estaba empezando a surgir en los últimos años, pero no existía un estándar en la implementación de la integración de ambas. Se encuentra una pequeña producción de artículos al respecto, cada uno con sus propias soluciones. Sin embargo, se ha conseguido realizar un estudio agrupando estos artículos y analizando las posibles estrategias que se pueden tomar para realizar un sistema de recomendación con redes siamesas.

A su vez, se han aunado y clarificado términos que a veces se utilizaban erróneamente, como el término *tripleta* y se ha establecido que las *Twin Neural Networks* y las *Triplet Networks* son tipos de SNN.

Por otro lado, se han creado una serie de experimentos reproducibles tanto por la aclaración del diseño del modelo como por el acceso al código por parte del lector <https://github.com/masternico97/Siamese-Recommender-Systems>, que permite el uso de algunas implementaciones de modelos de sistemas de recomendación con SNN tanto para estrategias basadas en contenido como para estrategias de filtrado colaborativo.

Por último, y gracias a estos experimentos, se ha podido observar que aunque en teoría la función de pérdida de entropía binaria cruzada debería funcionar peor frente a la *contrastive loss* en este tipo de problemas según el estado del arte, en algunos modelos los modelos que utilizan como función de pérdida la entropía binaria cruzada arrojan mejores resultados. Así como que estos modelos pueden arrojar mejores resultados frente a modelos más tradicionales de sistemas de recomendación.

5.2. Trabajo futuro

Realizado ahora el estudio y dando diferentes modelos ya creados de sistemas de recomendación con redes siamesas, las posibilidades de expansión son prácticamente ilimitadas.

Primero, gracias a esta investigación, cualquier persona que quiera tener una referencia sobre cómo integrar las redes siamesas junto a los sistemas de recomendación puede observar el estado del arte para decidir qué tipo de integración es la óptima (o la que más se aproxima) para su problema, basándose en los últimos avances e integraciones en la investigación. Así mismo se clarifican términos de diferentes artículos de investigación para aproximar a futuros lectores a este tipo de modelos.

Segundo, este trabajo puede resultar como una pequeña plantilla de diseño o de código para la implementación de futuras aproximaciones en otros problemas de recomendación. Esto hace que el inicio de la creación de estos sistemas sean más sencillos, pudiendo desarrollar evoluciones más complejas.

Tercero, así como puede resultar como plantilla de la que partir para un modelo más complejo, estos modelos puede servir como líneas base para otros sistemas de recomendación.

Finalmente, la investigación y experimentos de este proyecto se pueden expandir en un futuro cuando el número de artículos en el área sea mayor o tratando como experimentos otras arquitecturas de sistemas de recomendación con redes siamesas.

BIBLIOGRAFÍA

- [1] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Boston, MA: Springer US, 2011.
- [2] Y. Koren and R. Bell, *Advances in Collaborative Filtering*, pp. 145–186. Boston, MA: Springer US, 2011.
- [3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering,” 2017.
- [4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a “siamese” time delay neural network,” *Advances in neural information processing systems*, vol. 6, 1993.
- [5] P. Tino, L. Benuskova, and A. Sperduti, “Artificial neural network models,” in *Springer Handbook of Computational Intelligence*, pp. 455–471, Springer, 2015.
- [6] E. Hoffer and N. Ailon, “Deep metric learning using triplet network,” in *Similarity-Based Pattern Recognition*, Lecture Notes in Computer Science, (Cham), pp. 84–92, Springer International Publishing, 2015.
- [7] Z. Yang, Z. Su, Y. Yang, and G. Lin, “From recommendation to generation: A novel fashion clothing advising framework,” in *2018 7th International Conference on Digital Home (ICDH)*, pp. 180–186, IEEE, 2018.
- [8] H. Yuan, G. Liu, H. Li, and L. Wang, “Matching recommendations based on siamese network and metric learning,” in *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1–6, IEEE, 2018.
- [9] S. Maheshwary and H. Misra, “Matching resumes to jobs via deep siamese network,” in *Companion Proceedings of the The Web Conference 2018*, pp. 87–88, 2018.
- [10] D. T. Le, H. W. Lauw, and Y. Fang, “Modeling contemporaneous basket sequences with twin networks for next-item recommendation,” *IJCAI*, 2018.
- [11] D. Khattar, V. Kumar, S. Gupta, M. Gupta, and V. Varma, “Rare: A recurrent attentive recommendation engine for news aggregators.,” in *CIKM Workshops*, 2018.
- [12] R. Burke and M. Ramezani, *Matching Recommendation Technologies and Domains*, pp. 367–386. Boston, MA: Springer US, 2011.
- [13] S. Jaradat, N. Dokoochaki, H. J. C. Pampín, and R. Shirvany, *Fashion Recommender Systems*, pp. 1015–1055. New York, NY: Springer US, 2022.
- [14] A. Khatua and W. Nejdl, “Matching recruiters and jobseekers on twitter,” in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 266–269, IEEE, 2020.
- [15] M. Guo, N. Yan, X. Cui, S. H. Wu, U. Ahsan, R. West, and K. A. Jadda, “Deep learning-based online alternative product recommendations at scale,” *CoRR*, vol. abs/2104.07572, 2021.

- [16] M. Angelovska, S. Sheikholeslami, B. Dunn, and A. H. Payberah, "Siamese neural networks for detecting complementary products," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 65–70, 2021.
- [17] G. Gao, L. Liu, L. Wang, and Y. Zhang, "Fashion clothes matching scheme based on siamese network and autoencoder," *Multimedia Systems*, vol. 25, no. 6, pp. 593–602, 2019.
- [18] L. F. Polanía and S. Gupte, "Learning fashion compatibility across apparel categories for outfit recommendation," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 4489–4493, IEEE, 2019.
- [19] K. Chen, B. Liang, X. Ma, and M. Gu, "Learning audio embeddings with user listening data for content-based music recommendation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3015–3019, IEEE, 2021.
- [20] M. Pulis and J. Bajada, "Siamese neural networks for content-based cold-start music recommendation.," in *Fifteenth ACM Conference on Recommender Systems*, pp. 719–723, 2021.
- [21] R. S. Sreepada and B. K. Patra, "Mitigating long tail effect in recommendations using few shot learning technique," *Expert Systems with Applications*, vol. 140, p. 112887, 2020.
- [22] W. Yuan, P. Wang, M. Yuan, Y. Guo, and T. He, "N2one: Identifying coreference object among user generated content with siamese network," in *International Conference on Web Information Systems and Applications*, pp. 276–288, Springer, 2020.
- [23] Y. Yu, H. Tang, F. Wang, L. Wu, T. Qian, T. Sun, and Y. Xu, "Tulsn: siamese network for trajectory-user linking," in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2020.
- [24] R. Vijjali, D. Bhageria, A. Tamhane, M. TM, and J. Sathyanarayana, "Foodnet: Simplifying online food ordering with contextual food combos," in *5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, pp. 178–185, 2022.
- [25] D. Perera and R. Zimmermann, "Cngan: Generative adversarial networks for cross-network user preference generation for non-overlapped users," in *The World Wide Web Conference*, pp. 3144–3150, 2019.
- [26] J. Neve and R. McConville, "Imrec: Learning reciprocal preferences using images," in *Fourteenth ACM Conference on Recommender Systems*, pp. 170–179, 2020.
- [27] C. J. Holder, S. Ricketts, and B. Obara, "Convolutional networks for appearance-based recommendation and visualisation of mascara products," *Machine Vision and Applications*, vol. 31, no. 1, pp. 1–13, 2020.
- [28] A. Faroughi and P. Moradi, "Moocs recommender system with siamese neural network," in *2022 9th International and the 15th National Conference on E-Learning and E-Teaching (ICeLeT)*, pp. 1–6, IEEE, 2022.
- [29] Y. Zhao, M. Qiao, H. Wang, R. Zhang, D. Wang, K. Xu, and Q. Tan, "Tdfi: Two-stage deep learning framework for friendship inference via multi-source information," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 1981–1989, IEEE, 2019.

- [30] M. Ostendorff, T. Blume, T. Ruas, B. Gipp, and G. Rehm, “Specialized document embeddings for aspect-based similarity of research papers,” *arXiv preprint arXiv:2203.14541*, 2022.
- [31] T. Kumari, R. Sharma, and P. Bedi, “A contextual-bandit approach for multifaceted reciprocal recommendations in online dating,” *Journal of Intelligent Information Systems*, vol. 59, no. 3, pp. 705–731, 2022.
- [32] Z. Li, S. Li, L. Xue, and Y. Tian, “Semi-siamese network for content-based video relevance prediction,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, 2019.
- [33] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” *ACM Comput. Surv.*, vol. 52, no. 1, pp. 5:1–5:38, 2019.
- [34] D. Valcarce, A. Bellogín, J. Parapar, and P. Castells, “Assessing ranking metrics in top-n recommendation,” *Inf. Retr. J.*, vol. 23, no. 4, pp. 411–448, 2020.
- [35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. Software available from tensorflow.org.
- [36] F. Chollet *et al.*, “Keras,” 2015.
- [37] A. Rosebrock, “Siamese networks with keras, tensorflow, and deep learning,” Nov 2020.
- [38] Y. LeCun, C. Cortes, and C. Burges, “Mnist handwritten digit database,” *ATT Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>*, vol. 2, 2010.
- [39] V. Luhaniwal, “Building visual similarity based recommendation,” 2021.
- [40] V. Luhaniwal, “E-commerce product images,” 2021.
- [41] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A dataset for music analysis,” in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.
- [42] M. Defferrard, S. P. Mohanty, S. F. Carroll, and M. Salathé, “Learning to recognize musical genre from audio,” in *The 2018 Web Conference Companion*, ACM Press, 2018.
- [43] A. N. Nikolakopoulos, X. Ning, C. Desrosiers, and G. Karypis, “Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems,” in *Recommender Systems Handbook* (F. Ricci, L. Rokach, and B. Shapira, eds.), pp. 39–89, Springer US, 2022.
- [44] N. Saxena, V. Khosla, A. Venkataraman, N. Khosla, and V. Venkataraman, “Building an image-based shoe recommendation system,” 2013.
- [45] S. Golovkine, “Triplet loss for neural recommender systems.”
- [46] Cooper Union, “Anime recommendations database.”
- [47] F. M. Harper and J. A. Konstan, “The movielens datasets: History and context,” *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

ACRÓNIMOS

AUC Area Under the ROC Curve.

HR Hit Ratio.

MRR Mean Reciprocal Rank.

NDCG Normalized Discounted Cumulative Gain.

SNN Redes Siamesas.

APÉNDICES

CONSULTAS BASES DE DATOS

Para las búsquedas de artículos se han realizado dos consultas generalistas, para intentar extraer todos los artículos relacionados con la temática deseada de las bases de datos de Web of Science A.1 y de Scopus A.2.

```
(  
  TI="recommender systems" OR TS="recommender systems" OR  
  AB="recommender systems" OR AK="recommender systems" OR  
  TI="recommendation system" OR TS="recommendation system" OR  
  AB="recommendation system" OR AK="recommendation system" OR  
  TI="recommendation" OR TS="recommendation" OR AB="recommendation" OR  
  AK="recommendation"  
)  
AND  
(  
  TI="siamese network" OR TS="siamese network" OR AB="siamese network" OR  
  AK="siamese network" OR  
  TI="siamese neural network" OR TS="siamese neural network" OR AB="siamese  
  neural network" OR AK="siamese neural network" OR  
  TI="twin neural network" OR TS="twin neural network" OR AB="twin neural net-  
  work" OR AK="twin neural network"  
)
```

Cuadro A.1: Consulta en la base de datos de artículos Web of Science

```
(  
  TITLE-ABS-KEY("recommender systems") OR  
  TITLE-ABS-KEY("recommendation system") OR  
  TITLE-ABS-KEY("recommendation")  
)  
AND  
(  
  TITLE-ABS-KEY("siamese network") OR  
  TITLE-ABS-KEY("siamese neural network") OR  
  TITLE-ABS-KEY("twin neural network")  
)
```

Cuadro A.2: Consulta en la base de datos de artículos Scopus