UNIVERSITAT DE BARCELONA

Trabajo de final de grado

GRADO DE INGENIERÍA INFORMÁTICA

Facultad de Matemáticas e Informática
Universidad de Barcelona

Machine Unlearning: el arte de olvidar en la era de la Inteligencia Artificial

Autor: PABLO NORIEGA VÁZQUEZ

Director:	Julio Cezar Silveira Jacques-Junior
Realizado en:	Departamento de Matemáticas e Informática

Barcelona, 10 de junio de 2024

Resumen

El “machine unlearning” es un concepto que describe el proceso o conjunto de técnicas utilizadas para disminuir la influencia de ciertos datos en un modelo de aprendizaje automático que ya ha sido entrenado previamente. Este proceso es esencial en situaciones donde es necesario corregir errores en los datos de entrenamiento, proteger la privacidad de los individuos, eliminar datos que ya no son relevantes, y evitar que el modelo se sesgue o sobreajuste a los datos de entrenamiento.

Este trabajo se centra en profundizar en la técnica de “machine unlearning” en el contexto de predicción de edad, a partir de un conjunto de datos compuesto por imágenes faciales de diferentes grupos de edad, sexos y etnias.

Se implementan dos métodos sencillos para realizar “machine unlearning”. Además, se define una métrica de evaluación clara con el objetivo de medir tanto la precisión de las predicciones como la eficacia en la eliminación de datos de los modelos entrenados.

Finalmente, se realiza un análisis detallado del comportamiento de los modelos bajo diversas configuraciones generadas mediante las técnicas de “unlearning”. En este análisis se explican los resultados obtenidos, comparando la precisión de las predicciones y los comportamientos observados en los modelos.

Abstract

The term “machine unlearning” refers to a method or collection of methods used to mitigate the impact of specific data on a machine learning model that has already been trained. This procedure is crucial when it comes to removing data that is no longer relevant, safeguarding the privacy of individuals, fixing mistakes in the training set, and keeping the model from becoming biased or overfitting the training set.

In this work, the technique of “machine unlearning” is explored with respect to age prediction using a dataset consisting of facial photos belonging to various age groups, genders, and ethnicities.

We evaluate two simple methods for machine unlearning implementation. Furthermore, an explicit evaluation metric is provided to assess the performance of data removal from the training models as well as the prediction accuracy.

Ultimately, an extensive analysis of the behavior of the models in various configurations produced by the “unlearning” procedures is performed. This analysis explains the obtained results, comparing the prediction accuracy and the observed behaviors in the models.

Resum

El “machine unlearning” és un concepte que descriu el procés o conjunt de tècniques utilitzades per disminuir la influència de certes dades en un model d'aprenentatge automàtic que ja ha estat entrenat. Aquest procés és essencial en situacions on és necessari corregir errors en les dades d'entrenament, protegir la privacitat dels individus, eliminar dades que ja no són rellevants i evitar que el model esdevingui esbiaixat o sobreajustat a les dades d'entrenament.

Aquest treball es centra en aprofundir en la tècnica de “machine unlearning” en el context de predicció d'edat, a partir d'un conjunt de dades compost per imatges facials de diferents grups d'edat, sexes i ètnies.

S'implementan dos mètodes senzills per avaluar el “machine unlearning”. A més, es defineix una mètrica d'avaluació clara amb l'objectiu de mesurar tant la precisió de les prediccions com l'eficàcia en l'eliminació de dades dels models entrenats.

Finalment, es realitza una anàlisi detallat del comportament dels models sota diverses configuracions generades mitjançant les tècniques d'“unlearning”. En aquesta análisi s'expliquen els resultats obtinguts, comparant la precisió de les prediccions i els comportaments observats en els models.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi tutor, Julio Cezar Silveira Jacques-Junior, por su inestimable apoyo, dedicación y tiempo invertido en la realización de este trabajo. Su conocimiento y constante disposición para ofrecer ayuda han sido fundamentales para alcanzar los objetivos planteados. Sin su compromiso y mentoría, este proyecto no habría sido posible.

También quiero dar las gracias a mi familia por su apoyo incondicional, paciencia y motivación, que han sido de gran ayuda durante toda esta etapa.

Índice

1. Introducción

1.1. Definición del problema

El desafío del "machine unlearning" surge de la necesidad de abordar las implicaciones éticas, de seguridad y de privacidad en el desarrollo y la implementación de modelos de aprendizaje automático [1].

A medida que estos modelos se vuelven omnipresentes en una variedad de aplicaciones, desde la clasificación de imágenes hasta la generación de texto, surge una preocupación creciente sobre cómo manejar los datos utilizados para entrenarlos.

Según [2], la importancia del "machine unlearning" radica en varios aspectos cruciales:

- **Cumplimiento normativo y derechos individuales:** Legislaciones como el Reglamento General de Protección de Datos (GDPR) de la Unión Europea, otorgan a los individuos el derecho a solicitar la eliminación de sus datos personales. En este contexto, el "machine unlearning" se vuelve esencial para cumplir con estas regulaciones al permitir que los modelos olviden datos sensibles o personales [3].
- **Protección de la privacidad y seguridad de los datos:** Los modelos de aprendizaje automático pueden retener información sensible durante el entrenamiento, lo que plantea riesgos de privacidad si estos modelos son comprometidos o utilizados de manera indebida. El "machine unlearning"ofrece una solución al permitir la eliminación de estos datos sensibles, reduciendo así la exposición indebida.
- **Corrección de sesgos y actualización de datos:** Los modelos de aprendizaje automático pueden desarrollar sesgos no deseados o volverse obsoletos a medida que se entrenan con datos desactualizados [4]. El "machine unlearning" aborda estos problemas al permitir la eliminación o actualización de los datos problemáticos, promoviendo así la igualdad y la precisión en los resultados del modelo.
- **Responsabilidad ética y transparencia:** En un entorno donde la ética y la responsabilidad en la inteligencia artificial son temas cada vez más relevantes, el "machine unlearning" contribuye a la transparencia al permitir la explicación y justificación de cómo se manejan los datos utilizados para entrenar los modelos. Esto ayuda a garantizar que los modelos sean éticamente responsables y justos en su funcionamiento [2].
- **Costo y eficiencia en el reentrenamiento:** Es importante destacar que reentrenar un modelo desde cero puede ser extremadamente costoso en términos de recursos computacionales y tiempo [5]. El "machine unlearning" busca evitar este costo al permitir que los modelos olviden datos específicos sin la necesidad de un reentrenamiento completo, lo que resulta en un proceso más eficiente y rentable.

1.2. Objetivos del trabajo

1.2.1. Objetivo general

El objetivo principal de este trabajo de fin de grado es implementar y evaluar métodos de desaprendizaje o "unlearning" en el contexto del aprendizaje automático, centrándose en métricas de eficiencia y efectividad.

Estos métodos de desaprendizaje se aplicarán específicamente en un modelo de predicción de edad a partir de imágenes faciales, con el propósito de explorar cómo eliminar selectivamente información sensible o no deseada del modelo mientras se conserva el rendimiento en otras clases o muestras.

Este trabajo está inspirado en la competición de "machine unlearning" desarrollada por Google [6] en 2023. El objetivo consistía en implementar una técninca que permitiese a un modelo de "machine learning" entrenado para el reconocimiento de edad a partir de imágenes faciales, olvidar un conjunto de datos definido como "forget set".

1.2.2. Objetivos Específicos

- **Desarrollo del Método de Desaprendizaje:**
  - Investigar y revisar las técnicas existentes de desaprendizaje.
  - Implementar un método de desaprendizaje que sea capaz de mantener una alta precisión en las predicciones mientras olvida los datos deseados, utilizando a su vez un nivel bajo de recursos computacionales.
- **Evaluación de la Efectividad y Eficiencia del Método:**
  - Definir métricas de evaluación claras y precisas para medir la efectividad y la eficiencia del método de desaprendizaje propuesto.
  - Aplicar el método de desaprendizaje al modelo de reconocimiento de edad previamente entrenado y evaluar su capacidad para eliminar la información deseada mientras se conserva un rendimiento satisfactorio en las predicciones.
  - Comparar el rendimiento del modelo antes y después de aplicar el desaprendizaje utilizando las métricas definidas, destacando las mejoras en eficiencia y la capacidad de adaptación del modelo.
  - Analizar los resultados obtenidos y discutir las implicaciones del método de desaprendizaje implementado en el contexto de este trabajo.

1.3. Estructura de la Memoria

El trabajo se divide en cuatros secciones diferentes:

- **Trabajos Relacionados**

  En esta sección se presentan diversas técnicas de "unlearning" existentes. Se explican, ventajas, desventajas y aplicaciones. También se mencionan diferentes métricas para evaluar su eficacia.

- **Modelos Implementados**

  En esta sección se detalla la elección de los modelos que se utilizaron como base en los experimentos. Se explican los ajustes realizados en ellos y las diferentes configuraciones que se probaron para llegar al resultado final.

- **Experimentos y Resultados**

  En este apartado, inicialmente se detalla el problema a tratar, el reconocimiento de edad. A continuación, se explica el proceso seguido para obtener los datos finales, que han sido utilizados durante el entrenamiento de los diferentes modelos.

  Por último, se describen los diversos experimentos realizados durante el trabajo. Se explican los objetivos de cada uno, se muestran los datos obtenidos y se presentan las métricas utilizadas para comparar las predicciones de los modelos.

- **Conclusiones y trabajos futuros**

  En esta última sección se explican los resultados obtenidos y los principales hallazgos sobre los comportamientos observados de los modelos.

  A continuación, se discuten las limitaciones encontradas durante el trabajo y se proponen posibles enfoques para desarrollos futuros.

2. Trabajos Relacionados

En esta sección se describen trabajos relacionados que abordan el "unlearning" en el contexto del aprendizaje automático. Se analizan diversas estrategias y enfoques propuestos para eliminar información no deseada de modelos previamente entrenados, junto con distintas métricas para evaluar la efectividad de los resultados obtenidos.

El área de "unlearning" se puede dividir en dos enfoques principales: "exact unlearning" y "approximate unlearning" [7]. En el "exact unlearning", el modelo se ajusta de manera precisa para lograr la ausencia total de los datos que se desean olvidar, eliminando selectivamente la información específica de un modelo entrenado sin afectar negativamente otras capacidades del modelo. En cambio el "approximate unlearning" utiliza métodos más eficientes pero no garantiza la eliminación total de los datos de los usuarios. A continuación se presentan varios métodos basados en estos enfoques.

2.1. Naive retraining

Esta técnica consiste en eliminar el conjunto de datos que se desea olvidar de los datos de entrenamiento originales y volver a entrenar el modelo desde cero [8] [9]. Implica descartar completamente el modelo original y entrenar uno nuevo.

La principal ventaja de esta implementación consiste en la garantía de que cualquier influencia del dato que queremos olvidar se elimine del modelo, ya que no se utiliza en el nuevo proceso de entrenamiento.

2.1.1. Desventajas

El método de "naive retraining" presenta desventajas importantes que impactan en su eficiencia y practicidad en el aprendizaje automático:

- **Alto coste computacional.**

  El "naive retraining" implica reentrenar el modelo desde cero, lo que significa volver a ajustar todos los pesos y sesgos durante el entrenamiento. Esto requiere una considerable cantidad de tiempo y recursos computacionales, especialmente al trabajar con modelos complejos o grandes volúmenes de datos.

- **Dependencia de datos originales.**

  Este enfoque necesita acceso completo a los datos originales de entrenamiento en cada iteración de reentrenamiento. La dependencia constante de grandes conjuntos de datos puede ser una limitación, especialmente si los datos están restringidos o no son fácilmente accesibles.

2.1.2. Aplicaciones

Es esencial destacar que el "naive retraining" no constituye un método de "unlearning" en sí mismo, ya que no implica un proceso activo de eliminación de información del modelo. En su lugar, se utiliza como referencia para evaluar la efectividad de otros algoritmos de "unlearning".

Se presupone que después de aplicar un algoritmo con el fin de olvidar ciertos datos, los resultados deberían aproximarse a los obtenidos mediante el "naive retraining". Esto indica que el proceso de "unlearning" ha sido exitoso en la eliminación o modificación de la información deseada.

2.2. SISA

SISA(Sharding, Isolation, Slicing, and Aggregation), [8],[10],[11]. Es una estrategia avanzada para realizar "unlearning" en modelos de aprendizaje automático de manera eficiente y precisa. Pertenece al grupo de "exact unlearning". La metodología SISA se centra en fragmentar, aislar, segmentar y posteriormente utilizar submodelos entrenados en subconjuntos de datos específicos para obtener una predicción final.

2.2.1. Proceso de aprendizaje con SISA

Para lograr una implementación efectiva de esta técnica de "unlearning", es necesario durante el entrenamiento del modelo seguir una serie de pautas detalladas a continuación:

1. **Fragmentación (Sharding).**

   Se divide el conjunto de datos en varios fragmentos disjuntos llamados "shards", permitiendo que cada "shard" sea utilizado para entrenar un submodelo independiente.

2. **Aislamiento (Isolation).**

   Cada submodelo se entrena de manera aislada, sin intercambio de información entre ellos. Esto asegura que la influencia de cada punto de datos esté restringida al modelo que lo está utilizando.

3. **Segmentación (Slicing).**

   La segmentación se divide a su vez en diferentes pasos:

   - **Subdivisión en Slices**

     Dentro de cada "shard" (subconjunto de datos), los datos se subdividen en pequeños "subsets" disjuntos llamados "slices". Cada "slice" representa una porción específica de los datos dentro del "shard". La principal razón para crear esta subdivisión de datos es aumentar la granularidad

del control, y poder identificar de forma sencilla y precisa el conjunto de datos al que pertenece la información que se desea eliminar.

- **Entrenamiento Incremental**

  Los modelos constituyentes se entrenan de manera incremental utilizando estos "slices". En lugar de entrenar el modelo con todos los datos del "shard" de una sola vez, se realizan múltiples rondas de entrenamiento, agregando gradualmente más "slices" al proceso.

  Este procedimiento permite en primer lugar mejorar la estabilidad del modelo, ya que al entrenar de manera incremental permite que el modelo se ajuste gradualmente, lo que puede evitar saltos bruscos en los parámetros de aprendizaje.

  En segundo lugar, al entrenar el modelo de manera incremental, es más fácil identificar y eliminar la influencia de datos específicos cuando se requiere "unlearning".

  La práctica habitual es definir los conjuntos de entrenamiento y validación para cada "shard" de la siguiente manera:

  - **Conjunto de Entrenamiento**
    En cada "shard", se empieza a entrenar con uno o más "slices", y gradualmente se añaden más hasta completar el entrenamiento total del "shard".
  - **Conjunto de Validación**
    Con el fin de poder evaluar el rendimiento del modelo de manera objetiva, dentro del "shard" se reservan una parte de los "slices" como conjunto de validación. Este conjunto no se usa durante el entrenamiento, sino que se utiliza para evaluar la precisión y generalización del modelo después de cada ronda de entrenamiento con nuevos "slices".

- **Guardado de Parámetros**

  Antes de incluir un nuevo "slice" en el entrenamiento, los parámetros del modelo constituyente se guardan. Esto significa que se almacena el estado actual del modelo antes de cada iteración de entrenamiento con un nuevo "slice".

- **Reanudación del Entrenamiento**

  Si es necesario reanudar el entrenamiento desde un punto anterior, los parámetros guardados permiten comenzar desde el estado previo. Esto es útil si se necesita ajustar el modelo o realizar "unlearning" en un punto específico del conjunto de datos.

4. **Agregación (Aggregation).**

   Finalmente, los modelos constituyentes se combinan utilizando estrategias como el voto mayoritario para formar el modelo final.

2.2.2. Proceso de desaprendizaje con SISA

Cuando surge la necesidad de realizar "unlearning" en un punto de datos específico, se sigue un proceso estructurado. En primer lugar, se identifica el modelo constituyente y el segmento dentro del conjunto de datos que contiene el punto a ser olvidado. El reentrenamiento comienza desde el último estado de parámetros guardado antes de incluir el segmento que contiene el punto a ser olvidado. Solo los modelos afectados por el cambio necesitan ser reentrenados, lo cual resulta una minimización significativa de la carga computacional y del tiempo necesario para adaptar el modelo.

Este enfoque modular y escalable del "framework"[1] SISA permite un aprendizaje eficiente y una adaptación precisa en entornos donde se requiere "unlearning" selectivo de datos en modelos de aprendizaje automático. Comparado con el enfoque de "Naive retraining", SISA reduce de manera efectiva la carga computacional y el tiempo requerido para adaptar los modelos, lo cual representa una mejora sustancial en la eficiencia del proceso de "unlearning".

A pesar de que es una técnica con buenos resultados, SISA también presenta una gran desventaja, al dividir los datos en muchos subconjuntos, se pierde rendimiento general en comparación con un modelo entrenado con todos los datos.

2.2.3. Eficiencia

Para comprobar la mejora que introduce dicha técnica en comparación al "Naive retraining" se puede utilizar una métrica específica llamada "speed-up".

Esta se define como una medida para calcular la rapidez de SISA respecto al "Naive retraining. Se calcula como $speedup = (R + 1)S$, donde:

- R: Número de "slices" en los que se divide cada "shard".
- S: Indica el número total de "shards" en los que se divide el conjunto de datos completo.

En esta métrica, (R+1) representa el número total de configuraciones de modelos constituyentes para cada "shard", considerando todas las posibles combinaciones de "slices" (desde 0 hasta R "slices"). La combinación de un mayor número de "shards" S y "slices" R permite explorar diferentes configuraciones de modelos, lo que proporciona un gran grado de flexibilidad y adaptación al proceso de entrenamiento. Sin embargo, es importante encontrar un equilibrio entre la división del conjunto de datos y recursos computacionales de los que se dispone para asegurar un proceso eficiente y efectivo de "unlearning".

[1]Un framework es una estructura de soporte definida que facilita el desarrollo de software. Proporciona un conjunto de herramientas, bibliotecas y buenas prácticas que permiten a los desarrolladores construir aplicaciones de manera más eficiente y coherente.

2.3. Fine-Tuning

El "fine-tuning", es una técnica fundamental en el campo del aprendizaje automático que permite adaptar modelos de redes neuronales preentrenados para resolver tareas específicas con conjuntos de datos limitados [12], [13], [14]. Consiste en reutilizar arquitecturas previamente entrenadas en conjuntos de datos grandes, y ajustar o adaptar estas redes a nuevas tareas o dominios de interés. A continuación, se comentan las etapas de aprendizaje y desaprendizaje basadas en "fine-tuning".

2.3.1. Proceso de aprendizaje con Fine-Tuning

El proceso para entrenar un modelo usando esta técnica consta de diferentes pasos detallados a continuación:

1. **Reutilización de Redes Preentrenadas**

   Se selecciona una red neuronal preentrenada en un problema relacionado o similar al problema que se desea resolver. El objetivo de utilizar una red existente es que habrá aprendido representaciones útiles de características generales que pueden ser transferidas a la nueva tarea.

2. **Ajuste de Algunas Capas**

   Se realiza un ajuste de algunas capas de la red preentrenada para que sean más relevantes para la nueva tarea. Generalmente, las capas iniciales (por ejemplo, convolucionales en CNNs[2]) se mantienen fijas o "freezed" para retener las características generales aprendidas, mientras que las capas finales o "fully-connected" se modifican para adaptarse a las características específicas del nuevo problema.

3. **Continuación del Entrenamiento**

   Se continúa el entrenamiento del modelo utilizando el nuevo conjunto de datos específico. Durante este proceso, se optimizan los pesos de las capas ajustadas para minimizar la pérdida en la nueva tarea, aprovechando las características generales aprendidas previamente.

2.3.2. Ventajas del proceso de apredinzaje con Fine-Tuning

Este método para entrenar modelos presenta una serie de características que lo sitúan entre los favoritos en el contexto de "machine learning".

- **Ahorro de Tiempo y Recursos:** Permite aprovechar el conocimiento previo aprendido por modelos preentrenados, evitando el costoso proceso de entrenamiento desde cero.

[2]Una red neuronal convolucional (CNN, por sus siglas en inglés) es un tipo de red neuronal artificial utilizada principalmente para el reconocimiento y procesamiento de imágenes, debido a su capacidad para reconocer patrones en las imágenes.

- **Mejora del Rendimiento:** Al adaptar modelos preentrenados a tareas específicas, se pueden obtener resultados más rápidos y precisos, especialmente con conjuntos de datos limitados.
- **Transferencia de Conocimiento:** Facilita la transferencia de conocimiento entre tareas relacionadas, permitiendo el uso eficiente de modelos previamente entrenados en nuevos contextos.

2.3.3. De aprendizaje a desaprendizaje

El "fine-tuning", comúnmente usado para entrenar modelos de aprendizaje automático con nuevos datos o para tareas específicas, también puede aplicarse para realizar "unlearning" o eliminar información no deseada.

Cuando se aplica el "fine-tuning" para hacer "unlearning" en un modelo, se ajustan de manera selectiva las representaciones internas del modelo con el objetivo de reducir la influencia de datos o clases específicas que se desean eliminar.

En lugar de reentrenar el modelo desde cero, el "fine-tuning" permite modificar partes clave del modelo para adaptarlo a nuevas condiciones o requisitos, incluyendo la capacidad de "olvidar" cierta información no deseada.

Durante este proceso, el modelo va eliminando gradualmente la información no deseada. Sin embargo, en la práctica, los resultados pueden ser incompletos ya que el modelo no consigue eliminar completamente la información. Por ese motivo, este método generalmente está relacionado con el grupo de "aproximate unlearning".

Para la implementación de esta técnica el procedimiento habitual consiste en eliminar parte de los datos, que serán aquellos que deseamos eliminar del modelo, estos se definen como "forget set". Con los datos restantes, definidos como "retain set"se realiza el "fine-tuning".

Tal y como se menciona en la Sección 2.3.4, se pueden aplicar unos cambios en el "forget set" con el fin de mejorar los resultados en precisión y eficiencia.

El modelo de referencia o "baseline"[3] que se proporcionaba en el "challenge" comentado en la Sección 1.2.1 , se basa en esta estrategia de "fine-tuning".

2.3.4. Fine-Tuning añadiendo perturbaciones

Otra estrategia basada en "fine-tuning" consiste en agregar perturbaciones adecuadas al estado inicial del modelo, previamente a la aplicación del "fine-tuning". Dichas perturbaciones están diseñadas para influir en los parámetros críticos que modelan los datos que se desea eliminar, permitiendo al modelo ajustarse a la tarea de desaprendizaje [15], [16].

Al introducir perturbaciones específicas, el modelo puede desestabilizar las representaciones asociadas con los datos no deseados de manera deliberada, facilitando

[3]Modelo (baseline) basado en "fine-tuning" usado en [6]: `https://www.kaggle.com/code/eleni30fillou/run-unlearn-finetune`

así el rápido ajuste de sus parámetros hacia una nueva configuración que consiga eliminar la información no deseada, tal y cómo se explica en [17].

Este enfoque puede acelerar el proceso de "unlearning" al permitir que el modelo adapte rápidamente sus representaciones internas para cumplir con el objetivo de eliminar datos específicos. Sin embargo, definir la perturbación adecuada y aplicarla de forma correcta puede suponer un gran desafío. Algunos ejemplos de perturbaciones que se pueden introducir en el modelo son:

1. **Información de Fisher.**

   En el contexto del aprendizaje automático, la información de "Fisher" [18] es una medida que captura la sensibilidad de un modelo a los cambios en sus parámetros. Su objetivo es proporcionar una comprensión de cómo los diferentes parámetros del modelo afectan las predicciones sobre los datos.

   Al aplicar perturbaciones basadas en la información de "Fisher" en el estado inicial del modelo, podemos enfocarnos en los parámetros que contribuyen significativamente a las predicciones sobre los datos que queremos eliminar.

   Esto puede ayudar a mejorar la capacidad del modelo para desaprender de manera efectiva al iniciar el proceso de "fine-tuning".

2. **Valores de activación.**

   En redes neuronales, otra forma de evaluar la importancia de los parámetros es observando cómo se activan las neuronas en respuesta a la entrada. Maximizar la activación de una neurona con ciertos datos puede revelar su función específica [19].

   En el contexto del desaprendizaje, se identifican las neuronas que se activan con los datos que queremos olvidar ($\mathcal{D}_f$ o "forget set") y se ajustan los parámetros relacionados. En una CNN con L capas, se calculan las puntuaciones de activación promedio para cada muestra de entrenamiento y se registran en una tabla. Posteriormente, se promedian las activaciones sobre el conjunto restante ($\mathcal{D}_r$ o "retain set") y el conjunto a olvidar ($\mathcal{D}_f$ o "forget set").

   Dicho enfoque permite introducir perturbaciones en los parámetros más relevantes para los datos que queremos olvidar.

3. **Información de Gradiente.**

   Este método evalúa la importancia de cada canal[4] de salida según cómo una pequeña perturbación en dicho canal afecta la pérdida final del modelo. Esto permite identificar canales relevantes para el desaprendizaje, enfocándose en aquellos que contribuyen significativamente a la pérdida en el conjunto de datos a olvidar ($\mathcal{D}_f$).

[4]En las capas intermedias de una red neuronal, como en una capa convolucional, los canales de salida se refieren a resultados de aplicar filtros convolucionales a los datos de entrada. Cada filtro produce un canal de salida que resalta ciertas características de la entrada, como bordes, texturas, etc.

Esta técnica, conocida como "GradMask"[20], selecciona neuronas basadas en su impacto en la pérdida asociada al conjunto a olvidar, lo cual puede ser más eficiente computacionalmente que otras estrategias que requieren un cálculo más costoso de la activación de cada muestra.

4. **Etiquetas aleatorias.**

   Esta técnica asume que cuando el modelo no tiene acceso a ciertos datos, debe comportarse como si estuviera adivinando al azar al asignar etiquetas a las entradas [21]. Aunque puede parecer una estrategia inicialmente válida, no es apropiada en todos los casos. Por ejemplo, si se tienen dos secuencias duplicadas pero se desea olvidar una y retener la otra, un modelo entrenado con etiquetas aleatorias no debería adivinar al azar en esta situación.

   La convergencia en etiquetas aleatorias a menudo resulta en una marcada disminución en la utilidad y el rendimiento del modelo, limitando así su efectividad a un breve período de ajuste de pesos.

Cada técnica presenta una serie de ventajas y desventajas, relacionadas directamente con el modelo en el que se desea implementar el "unlearning" y otras características como por ejemplo los recursos computacionales disponibles. Por este motivo no se puede escoger una estrategia universal para añadir perturbaciones, y se debe analizar con detalle el problema a tratar con el fin de escoger la mejor opción.

2.4. Reinicio selectivo de parámetros con Fine-Tuning

Este método obtuvo el sexto lugar en el concurso de Google mencionado en la Sección 1.2.1. En comparación con modelos con puntuaciones similares o superiores, dicho enfoque resulta muy sencillo y computacionalmente eficiente. Es un modelo [6][5] de "unlearning" simple pero efectivo que consta de tres fases diferentes: un reinicio selectivo de parámetros, una "fase de calentamiento" y finalmente un proceso de "fine-tuning".

1. **Reinicio de parámetros del modelo**

   En primer lugar se resetean la primera y última capa del modelo original. Al reiniciar la primera capa, se produce una interrupción en el modelo de las representaciones ocultas que se encontraron con los datos iniciales. Esto provoca una disminución directa del rendimiento del modelo, facilitando el olvido de los datos previamente entrenados.

   El reinicio de la última capa permite al modelo hacer predicciones que se desvíen de sus patrones de salida originales. De esta manera los resultados obtienen una nueva distribución de salida.

[5] `https://www.kaggle.com/competitions/neurips-2023-machine-unlearning/discussion/458740`

2. **Fase de calentamiento**

   Posteriormente al reinicio de parámetros se incluye una fase de calentamiento cuyo propósito es preparar el modelo para la última fase. El objetivo de esta fase es transferir el conocimiento del modelo original a nuestro nuevo modelo, minimizando la divergencia de Kullback-Leibler (KL) entre las salidas de los dos modelos.

   La divergencia de Kullback-Leibler [22] es una medida de diferencia entre dos distribuciones de probabilidad. Se utiliza, en este caso, para intentar que los resultados del modelo de unlearning se asemejen lo máximo posible a los resultados del modelo original.

3. **Fine-Tuning**

   Por último se lleva a cabo una fase final de "fine-tuning" en la que se entrena el modelo solo con los datos que se desean conservar, el "retain set" o $\mathcal{D}_r$. Para ello se utilizan tres funciones de pérdida diferentes [23], [24].

   - Cross-entropy loss: Asegura la precisión del modelo en la tarea utilizando la etiqueta que indica la clase del dato.
   - Soft cross-entropy loss: compara predicciones suavizadas del modelo de "unlearning" con los valores del modelo original. Estas predicciones no son simplemente una etiqueta de clase única, sino una distribución de probabilidad sobre todas las clases posibles.
   - KL divergence loss: facilita una transferencia más efectiva del conocimiento del modelo original al modelo de desaprendizaje, capturando una información más amplia y refinando las predicciones del modelo de desaprendizaje en función de las distribuciones de probabilidad del modelo original.

   Al combinar estas técnicas, se logra una mayor robustez y eficiencia del modelo de desaprendizaje, lo que resulta en una retención efectiva del conocimiento del modelo original mientras se evita el sobreajuste a las etiquetas y se captura información más amplia y diversa.

Como se explica durante la sección, existen distintos modelos para el desaprendizaje automático. Este trabajo, analiza principalmente modelos que no requieran de grandes recursos computacionales y logren una eficiencia y rendimiento satisfactorios. Para explorar alternativas y obtener conocimiento sobre más técnicas, recomendamos consultar [8], [25].

2.5. Métricas de evaluación

En esta sección, se analizan algunas métricas de evaluación utilizadas para medir el desaprendizaje en el contexto del aprendizaje automático. Estas métricas son fundamentales para comprender y cuantificar la efectividad de las estrategias y

técnicas aplicadas para eliminar información no deseada. Su finalidad consiste en evaluar la capacidad de un modelo para olvidar datos específicos, así como para medir la retención de otras habilidades o conocimientos útiles durante el proceso de desaprendizaje.

Estas métricas se utilizan principalmente en tareas de clasificación, ya que la mayoría de los trabajos relacionados con este campo se enfocan en resolver problemas de clasificación [26]. Esta preferencia por la clasificación se debe a la amplia gama de aplicaciones prácticas que abarca y a la mayor facilidad para obtener datos etiquetados para la fase de entrenamiento.

2.5.1. Accuracy

La precisión o "accuracy" de un clasificador es una métrica común utilizada para evaluar el desaprendizaje en el contexto del aprendizaje automático. Dicha métrica se utiliza para medir la capacidad de un modelo para clasificar correctamente los datos después de haber sido ajustado para eliminar información no deseada [27].

Se suele evaluar en dos dos conjuntos de datos clave: en un conjunto de prueba ($\mathcal{D}_t$ o "test set") y el conjunto de datos a olvidar ("forget set" o $\mathcal{D}_f$).

1. **Accuracy en el conjunto de pruebas**

   Aquí, el "accuracy" se refiere a la capacidad del modelo al que se le ha aplicado "unlearning" para predecir correctamente los valores en el conjunto de prueba. Es decir, se mide si el rendimiento del modelo está a un nivel parecido que el entrenado desde cero. Se espera que el "accuracy" en el conjunto de prueba se mantenga alto o no disminuya significativamente después del desaprendizaje. Esto indica que el modelo aún puede generalizar bien, incluso después de haber eliminado cierta información.

2. **Accuracy en el forget set**

   El "forget set" representa específicamente los datos que se deseaban olvidar durante el proceso de desaprendizaje. Se espera que el "accuracy" en el conjunto de datos a olvidar o $\mathcal{D}_f$ sea similar al del modelo entrenado desde cero. Trabajos recientes [6], se esfuerzan por hacer que estas predicciones se acerquen lo máximo posible a las del modelo entrenado desde cero, ya que representa el "unlearning" ideal, donde los datos que se deseaban olvidar no han influido en absoluto en el modelo.

2.5.2. Membership Inference Attach (MIA)

El Membership Inference Attack (MIA) es una técnica que se utiliza para determinar si ciertos ejemplos fueron parte del conjunto de datos de entrenamiento de un modelo de aprendizaje automático. En otras palabras, el MIA intenta averiguar si un ejemplo dado fue utilizado para entrenar el modelo o no [28], [29]. En el contexto de los modelos de lenguaje de gran escala (LLMs), el MIA se utiliza para verificar

si ciertas secuencias o ejemplos han sido eliminados correctamente del conjunto de datos de entrenamiento del modelo [30]. Se logra evaluando cómo el modelo predice palabras individuales y comparando estas predicciones con las del conjunto de entrenamiento original.

El método Min-K Prob [31], que es una técnica común de MIA, se centra en identificar palabras inusuales o atípicas en ejemplos que no formaban parte del conjunto de entrenamiento original. Si se encuentran diferencias significativas en las predicciones entre ejemplos "miembros" (que estaban en el conjunto de entrenamiento) y "no miembros" (que no lo estaban), el MIA puede determinar si el modelo ha "olvidado" correctamente ciertos ejemplos. Si el modelo ha olvidado correctamente, se esperaría que las predicciones para los ejemplos "no miembros" fueran menos precisas o más inciertas en comparación con las predicciones para los ejemplos "miembros. Se debe a que los ejemplos "miembros" formaban parte del conjunto de entrenamiento original y el modelo debería haber aprendido a predecirlos bien. En cambio, los ejemplos "no miembros" no fueron utilizados en el entrenamiento y, por lo tanto, el modelo podría mostrar un rendimiento diferente al intentar predecirlos.

En resumen, el Membership Inference Attack es una herramienta para evaluar si el modelo puede distinguir entre ejemplos que formaban parte del conjunto de entrenamiento y aquellos que no lo eran, con la finalidad de verificar la efectividad del proceso de desaprendizaje.

2.5.3. Interclass Confusion Test

El "Interclass Confusion Test" consiste en una métrica diseñada para verificar si un modelo al que se le ha realizado "unlearning" retiene información sobre los datos específicos que se desean olvidar. A diferencia de otras métricas que se enfocan en la similitud entre modelos en términos de parámetros, este test se centra en la salida del modelo [32].

En este test, se eligen aleatoriamente muestras de dos clases distintas en los datos de entrenamiento y se intercambian sus etiquetas. Esto crea un conjunto de muestras "confundidas". Posteriomente, se entrena un nuevo modelo utilizando este conjunto junto con el resto de los datos originales. Posteriormente, se calcula una puntuación utilizando una matriz de confusión generada por el modelo de "unlearning". Un valor bajo indica un mejor desaprendizaje del modelo, lo que significa que el modelo de "unlearning" es menos propenso a retener información sobre los datos que se intentaron olvidar.

2.5.4. Feature Injection Test

El "Feature Injection Test" es una métrica que se utiliza para evaluar si un modelo, después de haber sido "desaprendido", ha ajustado correctamente los pesos relacionados con los datos que se eliminaron, en función de ciertas características de los datos [2]. La idea detrás de esta métrica es introducir una característica adicional para cada dato en el conjunto de datos a olvidar, de manera que esta característi-

ca sea igual a cero para el conjunto de datos restantes pero esté perfectamente correlacionada con las etiquetas del conjunto a olvidar.

Se espera que el peso asociado con esta característica adicional sea significativamente diferente de cero, reflejando su importancia para la clasificación. Sin embargo, después de que el modelo se haya vuelto a entrenar para olvidar ciertos datos, este peso asociado debería tender a cero nuevamente. La efectividad del proceso de "desaprendizaje" se mide observando la diferencia en este peso antes y después del proceso. Si la diferencia es notable y el peso se aproxima a cero después del desaprendizaje, lo cual indica que el modelo ha ajustado adecuadamente los pesos asociados con los datos que se deseaba olvidar.

Es importante mencionar que esta métrica solo es aplicable a modelos lineales o logísticos debido a la forma explícita en que gestionan los pesos de las características. Para modelos más complejos, inyectar una característica de manera efectiva puede resultar más desafiante y requerir enfoques específicos.

3. Modelos Implementados

Para ilustrar de manera práctica las aplicaciones y características de algunos modelos mencionados en la Sección 2, durante la realización del trabajo hemos implementado una serie de modelos con variaciones en sus parámetros y estructura.

El problema que se pretende resolver con estos modelos es la predicción de edad a partir de imágenes faciales, de manera similar al que se propone en el desafío de Google [6]. A continuación se explica con detalle los diferentes modelos que se han utilizado.

3.1. Modelo Original

Con el objetivo de evaluar diferentes métodos de "unlearning", inicialmente necesitábamos entrenar un modelo de "machine learning" capaz de predecir edades a partir de imágenes faciales. Las opciones disponibles para entrenar un modelo en este contexto son principalmente dos: entrenar un modelo desde cero o utilizar un modelo preentrenado en un contexto similar y luego aplicar un "fine-tuning" para adaptarlo a nuestros datos.

Debido a nuestras limitaciones de recursos (e.g., no tener acceso a una GPU potente), entrenar un modelo desde cero capaz de aprender representaciones y características complejas, no era una opción viable, debido al tiempo de ejecución prolongado y los posibles problemas de rendimiento en un entorno de ejecución limitado (p.ej., cuando se utiliza Google Colab[6]). Por lo tanto, si hubiéramos optado por entrenar nuestro propio modelo desde cero, habría sido necesario utilizar una arquitectura simple, con un número reducido de capas, lo que conllevaría a una disminución del rendimiento del modelo.

Como nuestro objetivo principal era enfocarnos en el "unlearning", decidimos utilizar una ResNet50 [33], que ya había sido entrenada con imágenes faciales, posteriormente cambiamos la cabeza de regresión del modelo para adaptarlo a nuestro problema (como se ilustra en la Figura 1), y por último realizamos un "fine-tuning" con nuestros datos para adaptarlo a la tarea de reconocimiento de edad.

Para asegurar resultados coherentes y confiables, entrenamos tres modelos distintos siguiendo el mismo procedimiento de "fine-tuning". Posteriormente utilizamos un promedio de las predicciones de los diferentes modelos para reducir el posible impacto de las inicializaciones aleatorias, garantizando una mayor estabilidad y fiabilidad en las predicciones.

3.2. Modelos de Unlearning

Como se menciona en la sección anterior, la limitación de recursos ha jugado un papel importante en el momento de elegir que modelos implementar. Por este motivo decidimos implementar dos modelos de "unlearning" basados en la técnica

[6]Google Colab: `https://colab.google`

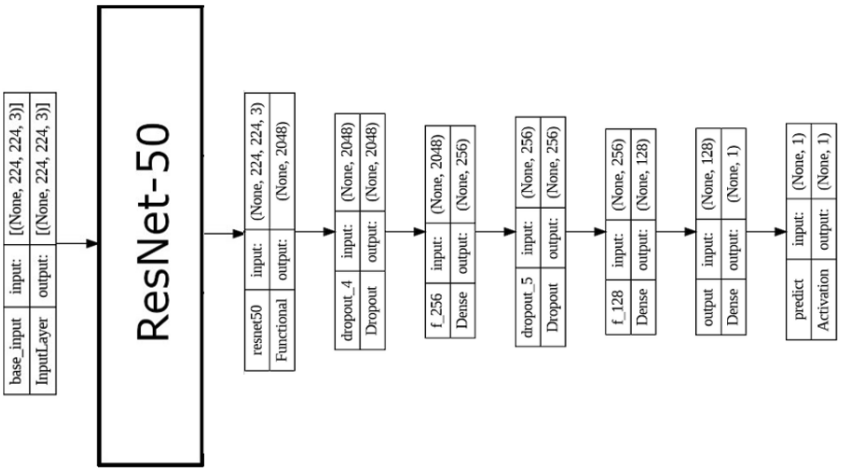

Figura 1: Modelo Original: ResNet50, con una cabeza de regresión modificada para predecir la edad.

de "fine-tuning" mencionada en la Sección 2.3. Además, cabe destacar que el área del desaprendizaje automático es bastante novedosa, y comenzar por este tipo de modelos más simples en lugar de otras técnicas de mayor complejidad, puede facilitar el proceso de entrenamiento e interpretación de resultados.

3.2.1. Fine-Tuning básico

El enfoque utilizado en este modelo se basa en una técnica sencilla de "fine-tuning". A continuación, se describen las diferentes fases del proceso:

1. **Entrenamiento del Modelo Original**: Primero, se entrena un modelo original utilizando todos los datos disponibles en el conjunto de entrenamiento. Este modelo debe alcanzar una precisión y un rendimiento adecuados antes de proceder al siguiente paso.

2. **Creación del Forget Set**: Una vez que el modelo ha sido entrenado, se definen los datos que el modelo debe olvidar. Este conjunto de datos se denomina "forget set" o $\mathcal{D}_f$.

3. **Creación del Retain Set**: Los datos del "forget set" se eliminan del conjunto de entrenamiento original. El conjunto de datos restante, después de esta eliminación, se llama "retain set" o $\mathcal{D}_r$.

4. **Reentrenamiento con el Retain Set**: Utilizando el "retain set", se vuelve a entrenar el modelo. Durante este reentrenamiento, las predicciones del modelo comenzarán a cambiar, ya que los datos del "forget set" ya no están presentes en el conjunto de entrenamiento.

Como resultado de este proceso, el modelo comenzará a introducir perturbaciones en las predicciones de los datos que se desean olvidar. A medida que el reentrena-

miento avanza, el modelo irá olvidando estos datos de forma incremental, ya que no tiene acceso a ellos durante el nuevo entrenamiento. Este enfoque permite eliminar efectivamente la influencia de ciertos datos en un modelo previamente entrenado, asegurando que el modelo se adapte a nueva configuración sin la información que se desea olvidar. Al ser una técnica básica de "unlearning", presenta dos limitaciones importantes:

- **Efectividad:** a pesar de que el modelo se va ajustando gradualmente a la ausencia de los datos que se pretenden olvidar, sigue conservando información sobre estos en sus representaciones internas. Por tanto, aunque el modelo sea capaz de reducir la influencia de la información a olvidar, no logra eliminar completamente su impacto en las predicciones.
- **Eficiencia:** este enfoque no resulta óptimo en términos de recursos computacionales y tiempo de entrenamiento, ya que puede requerir un elevado número de iteraciones para obtener resultados satisfactorios.

Los resultados que esperamos obtener para este modelo, deben de ser predicciones similares a las del modelo entrenado desde cero respecto a los datos que no se desean olvidar o "retain set" ($\mathcal{D}_r$), ya que no se introduce mucho ruido y por tanto la configuración interna no debería variar en gran cantidad. Por otro lado, cuando se se intentan predecir valores de datos que se desean eliminar o "forget set" ($\mathcal{D}_f$), los resultados obtenidos deberían de ser ligeramente peores a los del modelo original, ya que pese a no eliminar por completo el impacto de los datos, si que consigue mitigar su influencia.

En conclusión, la solución deseada sería aproximarse lo máximo posible a a los resultados obtenidos por el modelo original en el "retain set" o ($\mathcal{D}_r$), y a los obtenidos por el modelo entrenado desde cero en el "forget set" o ($\mathcal{D}_f$) ya que esto significaría que la influencia de los datos que se desean olvidar es mínima mientras se consigue mantener la precisión.

3.2.2. Fine-Tuning con etiquetas aleatorias

Para la creación de este modelo nos hemos basado en el enfoque anterior, pero añadiendo una serie de perturbaciones tal y como se explica en el apartado 2.3.4.

Para este modelo, probamos distintos enfoques para modificar las etiquetas del conjunto de datos que deseábamos olvidar ($\mathcal{D}_f$ o"forget set"), analizando el grado de impacto que tenían en el resultado de las predicciones. Las diversas técnincas se detallan a continuación:

- **Etiquetas aleatorias**

  En primer lugar probamos a substituir el valor de la etiqueta original por un número aleatorio entre el rango de edades contenidas en nuestra base de datos de imágenes.

Tras entrenar diversos modelos con "epochs" diferentes, concluimos que esta opción no era viable, ya que el impacto de las etiquetas modificadas era muy elevado y al final del reentreno las predicciones del modelo perdían demasiada precisión. Esto provocaba que pese a olvidar los datos deseados, el modelo carecía de rendimiento, ya que las predicciones no se asemejaban a los valores reales para el resto de datos que no se debían olvidar ("retain set" o $\mathcal{D}_r$) .

- **Etiquetas aleatorias con rango**

  La creación del modelo anterior nos llevo a un nuevo enfoque, ya que nos permitió comprobar que al introducir etiquetas aleatorias el modelo conseguía olvidar datos con mayor rapidez, siempre y cuando el ruido introducido no impactase de manera negativa en las predicciones. Por ese motivo decidimos establecer unos límites a la hora de elegir un nuevo valor aleatorio para las etiquetas originales. Es decir, en lugar de establecer cualquier valor del conjunto de datos, limitamos al modelo para que solo pudiera modificar el valor restando o sumando una cifra entre $[v_{min}, v_{max}]$, donde $v_{min} = 3$ y $v_{max} = 6$, para darle cierta flexibilidad al modelo.

  Eliminamos la opción de sumar o restar usando $v_{min} = 1$ y $v_{max} = 2$, ya que la variación de edad es mínima y su impacto no se vería reflejado al entrenar el modelo de "unlearning". Este hecho, fue observado durante la realización de los experimentos y fue clave para la definición de los hiperparámetros.

En resumen, observamos que al substituir las etiquetas originales por valores ligeramente modificados, se podría mejorar la eficiencia del modelo a la hora de olvidar datos, pero se debe de tener en cuenta el grado de ruido que se introduce, ya que la manera en la que modifiquemos los valores será beneficiosa para el "unlearning" o impactará negativamente en las predicciones reduciendo la eficiencia del modelo por completo. Además, la forma en que se evalúa el desaprendizaje también influye en esta decisión. Es decir, ¿Qué tan similares (o no) deberían ser las predicciones del modelo reentrenado en comparación con el modelo entrenado desde cero en el "forget set"?

En la siguiente sección, describimos y discutimos los experimentos realizados y los resultados que hemos obtenido usando estos dos enfoques (descritos en las secciones 3.2.1 y 3.2.2).

4. Experimentos y resultados

Esta sección detalla el problema que seleccionamos como caso de uso (el reconocimiento de edad), la métrica que definimos para evaluar los modelos, los experimentos realizados y los resultados que obtuvimos, con un análisis y discusión detallados.

El caso de uso que seleccionamos se detalla en la Sección 4.1. Luego, describimos el conjunto de datos utilizado en la Sección 4.2 y los pasos de preprocesamiento que utilizamos. La métrica de evaluación propuesta se describe en la Sección 4.3. En la Sección 4.4, describimos un experimento para evaluar el valor $alfa$ (α), un hiperparámetro utilizado por la métrica de evaluación. Luego, analizamos la influencia del número de "epochs" en el desaprendizaje en la Sección 4.5. Por último, evaluamos el caso en el que se cambia el conjunto de olvido ($\mathcal{D}_f$ o "forget set"), en la Sección 4.6.

Las Tablas 1 y 2 muestran las descripciones de los conjuntos y modelos, para ayudar al análisis de los experimentos[7].

Tabla 1: Abreviaturas de conjuntos de datos y sus descripciones.

Conjunto	**Descripción**
$\mathcal{D}_o$	Conjunto de entrenamiento original (con todas las muestras de entrenamiento)
$\mathcal{D}_f^j$	Forget set j: compuesto por las muestras que queremos olvidar, eliminadas del conjunto de entrenamiento original
$\mathcal{D}_r^j$	Retain set j: conjunto de entrenamiento restante ($\mathcal{D}_o$ sin el $\mathcal{D}_f^j$)
$\mathcal{D}_t$	Test set: conjunto de prueba independiente, que no se utilizó durante el entrenamiento

Tabla 2: Abreviaturas de modelos y sus descripciones.

Modelo	**Descripción**
$\mathcal{M}_O$	Modelo Original (entrenado con $\mathcal{D}_o$)
$\mathcal{M}_R$	Modelo entrenado desde cero, entrenado con $\mathcal{D}_r^j$
$\mathcal{M}_{FT}$	Modelo de unlearning 1 (fine tuning basico), entrenado con $\mathcal{D}_r^j$
$\mathcal{M}_{RND}$	Model de unlearning 2 (fine tuning con etiquetas aleatorias), entrenado con $\mathcal{D}_r^j$

4.1. Caso de Uso: reconocimiento automático de edad a partir de imágenes faciales

Inspirado en la competición de "machine unlearning" desarrollada por Google [6] en 2023, nuestro caso de uso también utiliza un escenario en el cual un predictor de

[7]El código utilizado durante la realización del trabajo se encuentra disponible en: `https://github.com/pablonoriega/TFG.git`

edad ha sido entrenado utilizando imágenes faciales de personas. Una vez se ha completado el entrenamiento, un conjunto específico de imágenes debe de ser borrado con el fin de proteger la privacidad o los derechos de los individuos involucrados.

La competición de Google [6] establece también los principales aspectos en los que se basarán a la hora de comparar los modelos entre si:

- **Tiempo de ejecución:** El modelo de "unlearning" debe de respetar un límite de tiempo respecto al que se tardaría volviendo a entrenar el modelo desde cero.
- **Eficacia:** Se mide el grado de "unlearning" que se ha conseguido en el modelo. Es decir, hasta qué punto el modelo es capaz de olvidar los datos que se pedían ($\mathcal{D}_f^j$ o "forget set").
- **Utilidad:** Una vez llevado a cabo el proceso de desaprendizaje, se comprueba si las predicciones del modelo sobre los datos no vistos durante el entrenamiento ($\mathcal{D}_t$ o "test set") son relativamente similares a las obtenidos por el modelo entrenado desde cero.

4.2. La base de datos UTK-FACES

A la hora de realizar nuestros experimentos, decidimos utilizar la base de datos UTK-FACES [34], siguiendo el protocolo detallado en [35] para definir los conjuntos de entrenamiento, validación y prueba. A continuación, se explican una serie de modificaciones que realizamos en el conjunto de datos para hacer frente a las limitaciones de recursos mencionadas anteriormente.

Esta base de datos está formada por más de 20.000 imágenes faciales con edades que varían entre 0 y 116 años. Cada muestra contiene anotaciones sobre edad, género y etnicidad. Las imágenes presentan una amplia diversidad en cuanto a poses, expresiones faciales, iluminación, resolución, entre otros factores. Por estas razones, UTK-FACES es un conjunto de datos ampliamente utilizado en diversos trabajos, como la detección de rostros, la estimación de edad y la localización de puntos de referencia. La distribución por edades de todo el conjunto de datos se ilustra en la Figura 2.

4.2.1. Reducción de datos

Como se menciona en la sección anterior, la base de datos inicial contaba con más de 20,000 muestras. Debido a nuestros recursos limitados al utilizar la herramienta pública Colab de Google, tuvimos que reducir este número, ya que nos era imposible entrenar un modelo de "machine learning" con este volumen de datos. Por ello, decidimos reducir el conjunto de datos y seleccionar el 30 % de las muestras, manteniendo la distribución de edades del conjunto original. La distribución de muestras de la base de datos reducida se ilustra en la Figura 3.

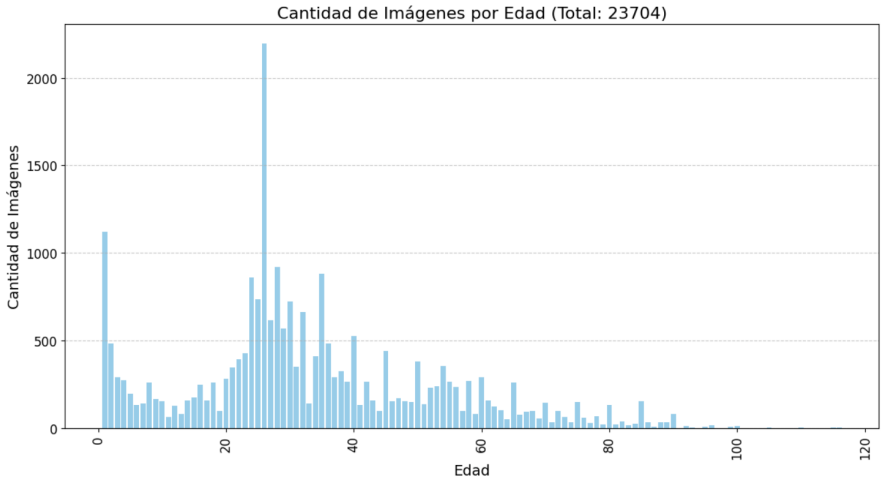

Figura 2: Distribución de muestras en la base de datos UTK [34].

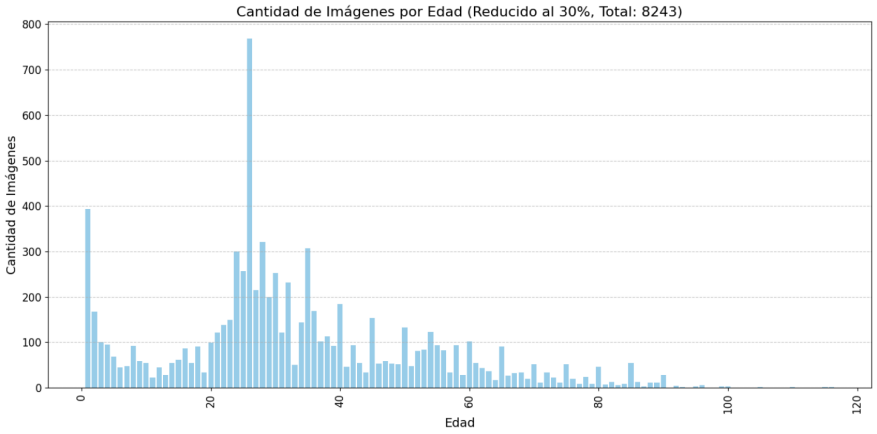

Figura 3: Distribución de muestras en la base de datos UTK reducida al 30 %.

Una vez reducido el número de imágenes, analizamos diversos aspectos del conjunto y decidimos enfocarnos únicamente en las imágenes dentro del rango de edad de 20 a 60 años. De esta manera, tendríamos un conjunto de datos reducido para trabajar, con un número relativamente razonable de muestras en los diferentes rangos de edad.

Algunas imágenes que estaban dentro del rango de edades de 20 a 60 años, pero que presentaban defectos como mal enfoque, baja resolución o problemas de iluminación, también fueron eliminadas. La distribución de muestras de la base de datos reducida, que contiene individuos con edades comprendidas entre 20 y 60 años, se ilustra en la Figura 4.

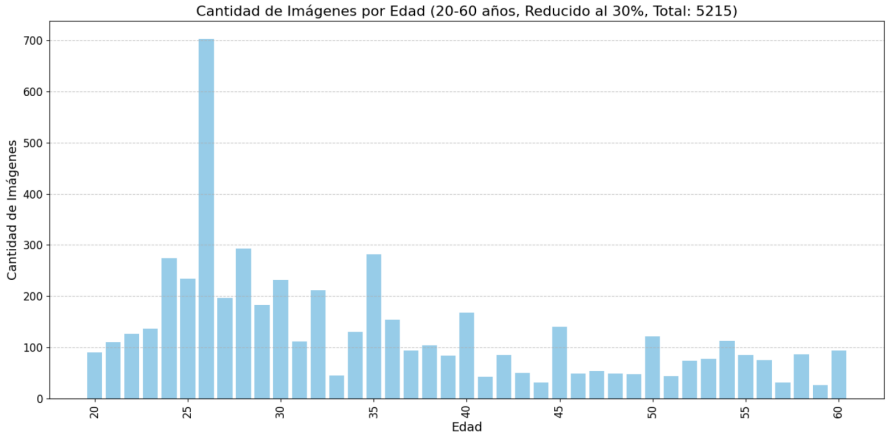

Figura 4: Distribución de muestras en la base de datos UTK reducida al 30 % y muestras seleccionadas entre 20 y 60 años.

Esta selección nos permitió trabajar con un conjunto de datos más manejable, adecuado para nuestros objetivos.

4.2.2. División en entrenamiento, validación y prueba

En cuanto a los datos de entrenamiento, los dividimos según los grupos que se detallan a continuación.

- **Conjunto de entrenamiento**

  Se utiliza con la finalidad de enseñar al modelo a hacer predicciones. Durante el entrenamiento el modelo ajusta los parámetros internos según las etiquetas de los datos, de esta manera consigue aprender patrones y características entre los datos de entrada y salida. Para este conjunto utilizamos el 60 % de los datos del conjunto, seleccionado aleatoriamente.

- **Conjunto de validación**

  Se utiliza durante el entrenamiento para ajustar los parámetros del modelo, como la tasa de aprendizaje o la regularización. No se usa directamente para entrenar el modelo, pero es de vital importancia para mejorar el rendimiento

del modelo al ajustar sus configuraciones externas. Para este conjunto hemos utilizado un 20 % de los datos del conjunto, seleccionado aleatoriamente.

- **Conjunto de prueba**

  Se reserva para evaluar el rendimiento del modelo al finalizar el proceso de entrenamiento. Estos datos no se utilizan durante el entrenamiento, lo que garantiza que el modelo no haya visto esos datos anteriormente. Nos da un resultado de cómo se comportará el modelo al hacer predicciones en datos nuevos. Para este conjunto hemos utilizado un 20 % de los datos del conjunto, seleccionado aleatoriamente.

La distribución de los conjuntos de entrenamiento, validación y prueba se ilustra en la Figura 5 y Figura 6.

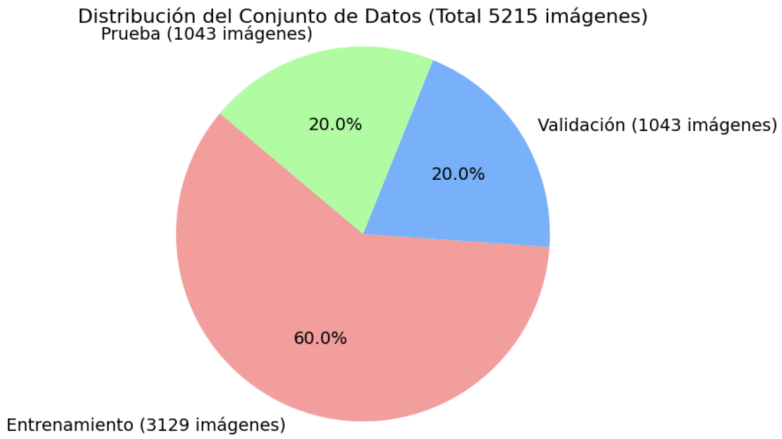

Figura 5: Distribución de datos en entrenamiento, validación y prueba.

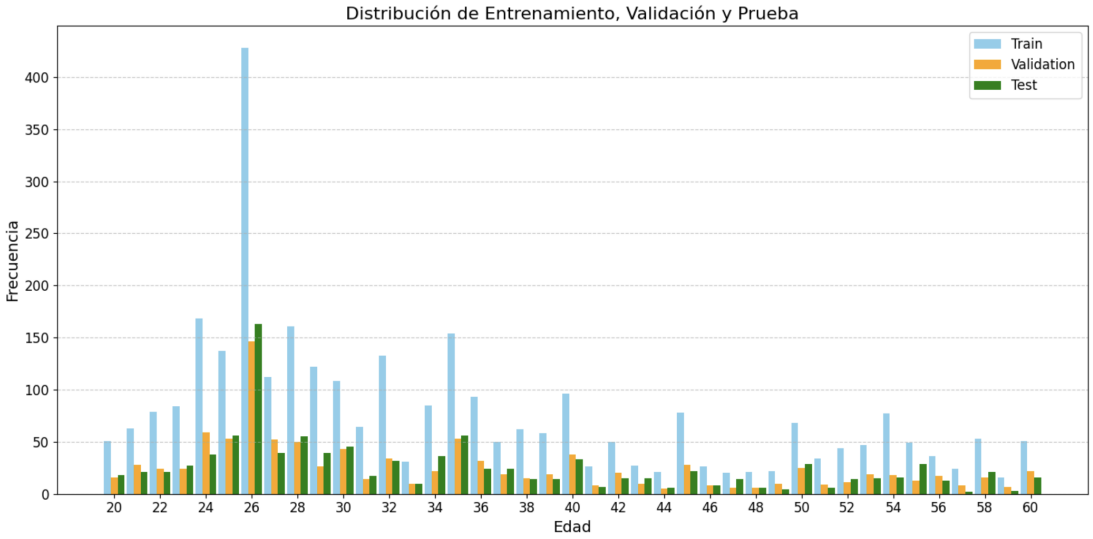

Figura 6: Distribución de datos en entrenamiento, validación y prueba.

4.2.3. Aumento de datos

Tras completar el proceso de reducir la base de datos, el conjunto de muestras ya era apto para ser utilizado en tareas de entrenamiento de modelos de "machine learning", adaptándose a nuestros limitados recursos. Sin embargo, los grupos de edad aún presentaban un desequilibrio significativo, ya que existía una gran disparidad en el número de muestras para diferentes años. Por este motivo decidimos aplicar una técnica ampliamente utilizada para equilibrar bases de datos, llamada "data augmentation".

Esta técnica implica seleccionar los grupos de la base de datos con menos muestras y añadir imágenes sintéticas, manteniendo la consistencia de las etiquetas. Estas nuevas imágenes se crearon mediante la aplicación de las siguientes transformaciones:

- **Rotación:** Rotación de la imagen en un rango de hasta 10 grados.
- **Desplazamiento Horizontal:** Desplazamiento horizontal de la imagen hasta un 10 % del ancho de la imagen.
- **Desplazamiento Vertical:** Desplazamiento vertical de la imagen hasta un 10 % de la altura de la imagen.
- **Zoom:** Zoom en la imagen hasta un 20 % (0.2).
- **Flip:** Volteo horizontal de la imagen.

En ninguna de las transformaciones aplicadas, la imagen sufre grandes cambios, asegurando que las muestras de cada edad en la base de datos no se alteren significativamente.

El objetivo del "data augmentation" es aumentar la diversidad de los datos disponibles para el entrenamiento del modelo y equilibrar la distribución de muestras. Esto puede favorecer su capacidad para generalizar patrones y hacer predicciones más precisas para diferentes intervalos de edad.

Debido a restricciones de recursos, limitamos el máximo de muestras generadas con data "augmentation" a 150 para cada edad. Por lo tanto, esta técnica no se aplicó a las edades que ya tenían más de 150 muestras en la base de datos. Esto asegura que no se introduzcan cambios significativos en las muestras de las edades que ya estaban bien representadas en el conjunto de datos original.

En la Figura 7 se muestra la distribución de los datos de entrenamiento después de aplicar este proceso.

Además de mejorar el desequilibrio de clases, el data "augmentation" puede ayudar a prevenir el "sobreajuste" del modelo al proporcionar una mayor variedad de ejemplos durante el entrenamiento. Aplicar esta técnica nos permitió mejorar la distribución nuestro conjunto de datos y mitigar el desequilibrio entre los grupos de edad. La distribución final de muestras que utilizamos para entrenar el modelo se ilustra en la Figura 8.

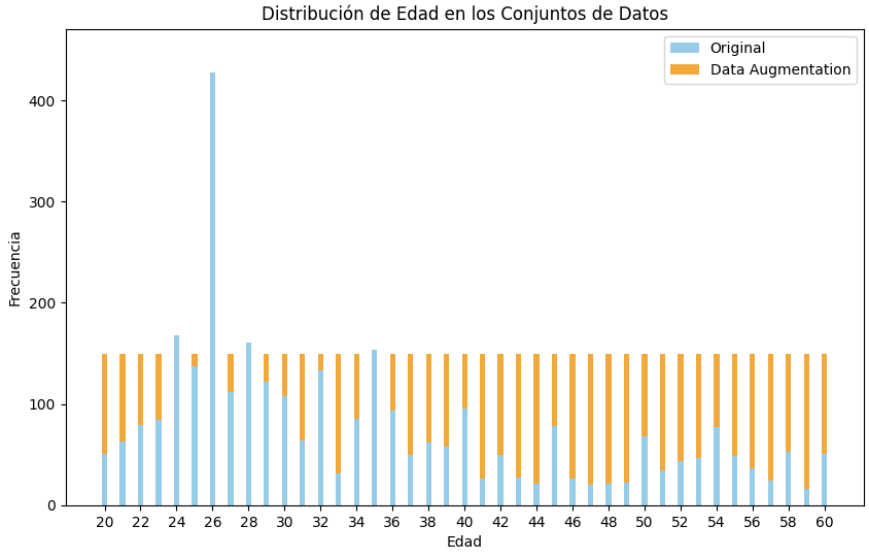

Figura 7: Distribuciones del conjunto de entrenamiento después de aplicar "data augmentation".

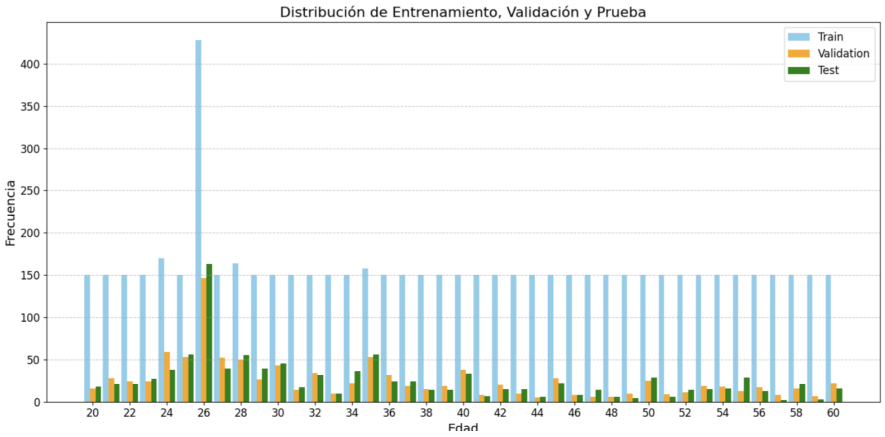

Figura 8: Distribuciones de los conjunto de entrenamiento, validación y prueba después de aplicar "data augmentation" en el "train".

4.3. Métrica de evaluación

Para analizar el rendimiento de los modelos, hemos desarrollado dos métricas que comparan el modelo evaluado con el modelo entrenado desde cero $\mathcal{M}_R$. Estas métricas nos permiten obtener resultados sobre la capacidad del modelo para olvidar información (evaluados en el "forget set" $\mathcal{D}_f^j$) y su precisión con respecto a los datos no vistos durante el periodo de entrenamiento (evaluada en el "test set" $\mathcal{D}_t$)

En relación con los datos a ser olvidados, buscamos un modelo con una capacidad lo más cercana posible a la del modelo entrenado desde cero, ya que este último representa el "unlearning" perfecto. En cuanto a la precisión con respecto a datos no vistos durante el periodo de entrenamiento ("test set" $\mathcal{D}_t$), buscamos un modelo que mejore la precisión lo máximo posible, ya que esto se relaciona con la capacidad de generalización del modelo. Para evaluar estos dos puntos, definimos "Utility" y "Forgetting", que se describen a continuación.

4.3.1. Utility

La "utility" o utilidad del modelo se centra en evaluar su capacidad de precisión respecto al "test set" o $\mathcal{D}_t$. Esta medida proporciona una comprensión de si el modelo reentrenado sigue siendo capaz de realizar predicciones significativas y coherentes después de haber sido sometido al proceso de desaprendizaje.

Al analizar la utilidad del modelo, podemos obtener información valiosa sobre su capacidad para adaptarse a los cambios en los datos y mantener su eficacia en la tarea de predicción de edad.

Se calcula utilizando como referencia la precisión (o error) dada por el modelo entrenado desde cero (sin el "forget set" $\mathcal{D}_f^j$). De esta forma podemos saber si el modelo evaluado mejoró o no la precisión del modelo de referencia, siendo este nuevo modelo el modelo original o el modelo de "unlearning".

Para ello se calcula el MAE[8] (Mean Average Error) en el conjunto de prueba, para el modelo de referencia (entrenado desde cero o $\mathcal{M}_R$) y el modelo que se está evaluando. Una vez calculado estos valores, definimos "Utility" como se muestra en la Ecuación 4.1.

$$Utility = \frac{M_r}{M_e}, \tag{4.1}$$

donde

- M_r se corresponde al MAE de las predicciones que genera el modelo entrenado desde cero $\mathcal{M}_R$, en el conjunto de pruebas.
- M_e se corresponde al MAE de las predicciones que genera el modelo el cual se quiere evaluar, en el conjunto de pruebas.

[8]El MAE, o Mean Absolute Error (Error Absoluto Medio), es una métrica que calcula el promedio de las diferencias absolutas entre las predicciones y los valores reales. Se define como: $\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$, donde y_i son los valores reales y $\hat{y}_i$ son las predicciones.

Si el resultado obtenido en esta ecuación es superior a 1, significará que el modelo que está siendo evaluado tiene una mayor precisión al entrenado desde cero. Por otro lado, valores inferiores a cero significan que el modelo evaluado tiene una precisión menor, en comparación con el modelo de referencia.

4.3.2. Forgetting

Esta métrica pretende analizar la capacidad del modelo para desaprender, es decir, su habilidad para olvidar una parte de los datos iniciales que se utilizaron para entrenar. A continuación se detalla cómo se calcula.

Para cada muestra i en el "forget set" $\mathcal{D}_f^j$, se calcula un valor f_i, dado por la Ecuación 4.2.

$$f_i = \frac{1}{\exp(E_i/\alpha)}, \tag{4.2}$$

donde

$$E_i = |r_i - e_i|, \tag{4.3}$$

En esta ecuación:

- r_i es el error absoluto dado por el modelo reentrenado desde cero en comparación con el valor esperado ("ground truth") para la muestra i.
- e_i es el error absoluto dado por el modelo evaluado (original o "unlearning") en comparación con el valor esperado ("ground truth") para la misma muestra i.
- Alfa (α) es un hiperparámetro que proporciona cierta flexibilidad a la métrica, como se detalla más adelante.

Es decir, E_i nos indica qué tan similar es la predicción del modelo evaluado (original o "unlearning") en comparación con la predicción dada por el modelo re-entrenado desde cero (sin el conjunto de datos a olvidar). Al calcular f_i para todas las predicciones de ambos modelos y hacer el promedio, podemos obtener un valor que nos permita evaluar la similitud general con respecto a lo que llamamos "forgetting".

$$Forgetting = \frac{1}{n}\sum_{i=1}^{n} f_i, \tag{4.4}$$

donde n es el número de muestras del "forget set" $\mathcal{D}_f^j$.

Cuanto más elevado sea el valor del "Forgetting", más similares serán las predicciones del modelo evaluado a las del modelo entrenado desde cero (sin los datos a olvidar). Por tanto, un valor alto representa una mayor capacidad para eliminar datos del modelo.

El parámetro α en la Ecuación 4.2 se emplea para determinar el nivel de penalización aplicado a los modelos cuando cometen errores en sus predicciones. Un valor bajo de α implica una penalización severa por los errores. Por otro lado, un valor

alto de α conlleva una penalización menos estricta, dando cierta flexibilidad a los modelos para cometer pequeños errores.

Por este motivo, es necesario encontrar un equilibrio adecuado en el valor de α que no penalice excesivamente los errores ni los permita en exceso. En la búsqueda de este equilibrio, nos esforzamos por establecer un punto medio que proporcione una penalización justa y proporcional a la magnitud de los errores cometidos por los modelos. De esta manera, podemos asegurar que la métrica resultante refleje la capacidad del modelo para adaptarse y aprender de manera efectiva, sin ser demasiado indulgentes ni demasiado estrictos al evaluar los resultados. A continuación, en la Figura 9 muestra diversos experimentos que llevamos a cabo para establecer el valor de α.

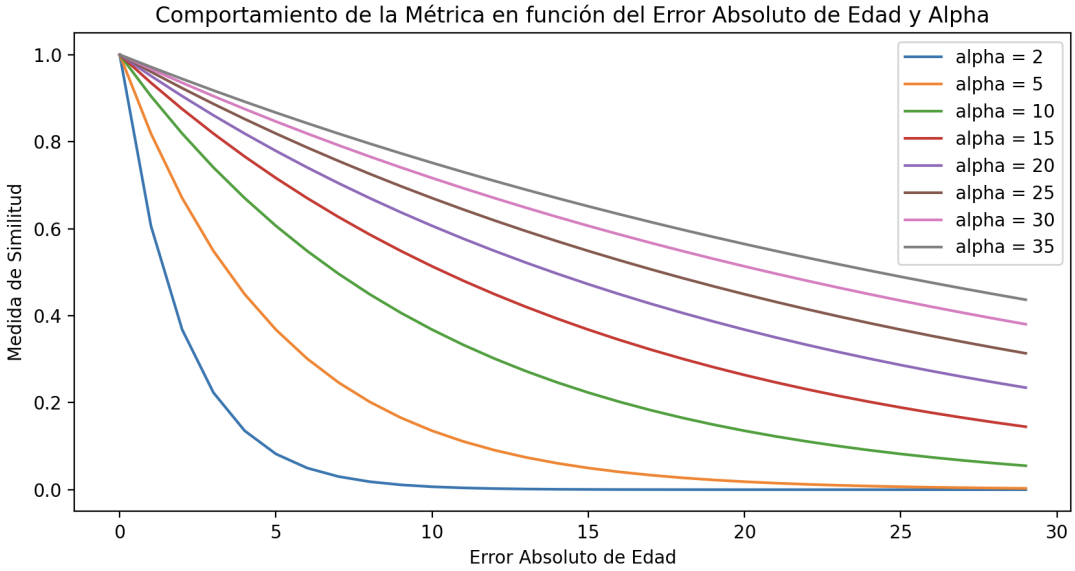

Figura 9: Relación entre similitud de modelos y penalización de errores.

4.3.3. Combining Utility and Forgetting

La métrica final (Ecuación 4.5) se obtiene multiplicando la "Utility" por el "Forgetting". De esta manera, usamos el valor de "Forgetting" para ponderar de alguna manera la "Utility".

$$F = Utility * Forgetting. \tag{4.5}$$

"Utility", se utiliza para evaluar el rendimiento del modelo en el "test set" o $\mathcal{D}_t$, respecto al modelo entrenado desde cero. En este contexto, un valor alto representará que nuestro modelo sigue haciendo predicciones similares a las del modelo original o $\mathcal{M}_O$, por lo que el proceso de desaprendizaje no habrá tenido un impacto negativo en su eficacia. Por otro lado, un valor bajo de "utility" indica que el modelo ha perdido precisión en sus predicciones y ya no se ajusta correctamente a los valores reales, lo que provoca una pérdida significativa de utilidad.

"Forgetting", tiene la finalidad de medir la capacidad del modelo para olvidar unos determinados datos. En este caso, un valor alto indicará que el modelo hace predicciones similares a las del modelo entrenado desde cero (sin los datos que se desean olvidar), lo que sugiere que el proceso de desaprendizaje habrá tenido éxito. Por otra parte, un valor bajo significará que las predicciones del modelo no se ajustan a las del modelo entrenado desde cero. Esto puede deberse a que los datos no se han olvidado adecuadamente, o bien a que se ha introducido demasiado ruido en el modelo, lo que ha alterado significativamente las predicciones.

Como resultado, el modelo de "unlearning" que logre obtener el valor más alto para F indicará que mantiene un rendimiento similar o por encima del modelo reentrenado desde cero $\mathcal{M}_R$, al mismo tiempo que consigue olvidar los datos de manera efectiva. Este hecho sugiere que el modelo ha logrado preservar su eficacia original mientras se adapta a nuevas circunstancias o cambios en los datos de entrada. En otras palabras, demuestra una buena capacidad para retener la información relevante y deshacerse de la información obsoleta o no deseada, lo que lo convierte en una herramienta valiosa para la adaptación a diferentes escenarios.

4.4. Experimento 1: creando los "forget sets" y analizando el valor alfa

Como se mencionó en la Sección 4.3.2, para evaluar los resultados es fundamental encontrar un valor adecuado de α. Por esta razón, llevamos a cabo una serie de pruebas para seleccionar el valor más apropiado

Con la finalidad de poder evaluar los resultados de "forgetting" de una manera adecuada respecto al valor de α , decidimos separar nuestra base de datos en grupos de ocho años, lo que generó la siguiente distribución.

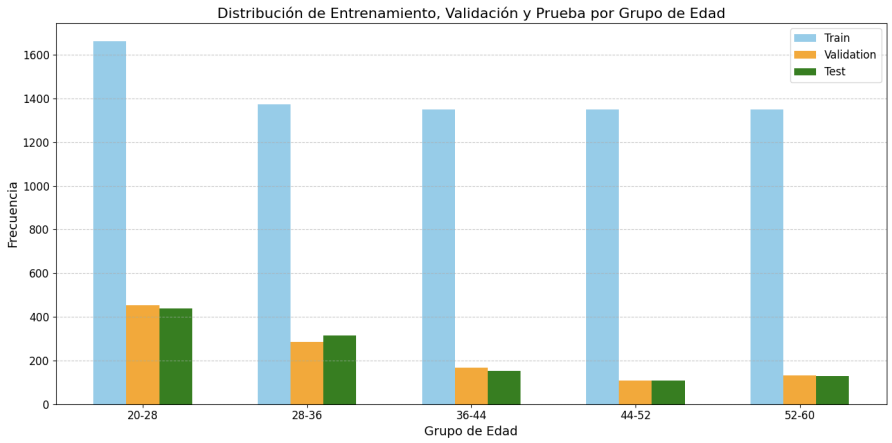

Figura 10: Distribución de los cinco grupos de edades.

Para este experimento, diseñamos cinco conjuntos de datos utilizando los grupos mostrados en la Figura 10. Cada conjunto estaba compuesto por cuatro de estos grupos, dejando uno fuera en cada caso. Esto nos permitió crear nuestros "forget sets" ($\mathcal{D}_f^j$), es decir, el grupo de datos que se desean eliminar del modelo (desde

$j = 0$ hasta $j = 4$), y nuestros "retain sets" ($\mathcal{D}_r^j$), es decir, el grupo de datos que se utilizarán para el entrenamiento de los modelos de "unlearning" o para el modelo entrenado desde cero.

Posteriormente, entrenamos diferentes versiones de los modelos (original, "fine-tuning" y "fine-tuning" con etiquetas aleatorias) explicados en la Sección 3, utilizando estos conjuntos de datos modificados. Esto nos permitió analizar el comportamiento de los modelos al intentar olvidar los datos específicos de cada grupo, evaluando así su capacidad para eliminar información de manera efectiva.

En este experimento, definimos las muestras de entrenamiento y validación de la primera categoría que se muestra en la figura 10 (edades de 20 a 28 años) como conjunto de olvido ("forget set" o $\mathcal{D}_f^0$), y las muestras de entrenamiento/validación restantes como conjunto de retención ("retain set" o $\mathcal{D}_r^0$). Los resultados presentados en la Tabla 3 representan el valor de "forgetting" (Ecuación 4.4) que obtiene cada modelo, cuando alfa cambia.

Es necesario destacar que, para aumentar la robustez del análisis, los valores obtenidos en los experimentos se han calculado a partir de N=3 ejecuciones (valor promedio).

Tabla 3: Valores de Forgetting (Ecuación 4.4) para diferentes valores de α y modelos.

Valor de α	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$
2	0,21	0,26	0,14
5	0,38	0,51	0,23
10	0,60	0,69	0,41
15	0,70	**0,78**	0,50
20	0,76	0,83	0,57
25	0,80	0,86	0,63
30	0,83	0,88	0,67
35	0,85	0,89	0,71

Como se puede observar en los valores de la Tabla 3, los valores de "forgetting" aumentan a medida que aumenta el valor de α se incrementa. Este comportamiento se explica en la Sección 4.3.2, donde se aclara que un valor bajo de α penaliza con más severidad los errores en las predicciones que comete el modelo, por lo que al aumentar α este error se penaliza con menos severidad.

No obstante, utilizar un valor elevado de α puede generar un desajuste en la métrica de evaluación, ya que significaría otorgar un alto grado de similitud a modelos cuyas predicciones no se ajustan a los valores esperados. Por ese motivo, optamos por emplear un valor intermedio de α, que no penalizara excesivamente, y al mismo tiempo, nos permitiese obtener valores de similitud apropiados. Como resultado, seleccionamos $\alpha = 15$ para todos los experimentos futuros.

La tabla 3 también muestra muestra (resaltado para $\alpha = 15$) que el modelo que genera predicciones más similares al entrenado desde cero en términos de utilidad/olvido (según nuestra métrica) es el "fine-tunning" $\mathcal{M}_{FT}$, seguido del modelo

original $\mathcal{M}_O$, y luego el "fine-tunning" con etiquetas aleatorias $\mathcal{M}_{RND}$. También es interesante mencionar que esta clasificación es consistente para todos los valores de alfa mostrados en la Tabla 3. Basándonos en los resultados de este experimento, mantendremos $\alpha = 15$ en los experimentos para darle cierta flexibilidad a los modelos.

4.5. Experimento 2: analizando el número de epochs

En el contexto de los modelos de "machine learning", una "epoch" es una de las etapas fundamentales del proceso de entrenamiento. A continuación se explica de manera detallada en que consiste.

Para entrenar un modelo es necesario contar con los datos de entrenamiento, estos son los que se utilizan para encontrar representaciones y características y poder evaluar las predicciones. En nuestro caso estos datos de entrenamiento están formados por imágenes faciales de personas etiquetadas con su edad.

Dado que procesar todo el conjunto de datos a la vez es muy costoso en términos de memoria y tiempo de computación, los datos se dividen en subconjuntos llamados "batches" o lotes, que están formados por un numero definido de muestras.

$$Nbatches = \frac{Ndatos}{BatchSize}, \tag{4.6}$$

donde

- $Nbatches$ = Número de batches total.
- $Ndatos$ = Número de datos total del conjunto de entrenamiento.
- $BatchSize$ = Tamaño de datos que contendrá cada batch.

Una "epoch" se define como un ciclo completo a través de todo el conjunto de datos de entrenamiento. Es decir, en una "epoch", el modelo procesa cada dato de entrenamiento una vez. Dentro de esta, el modelo se entrena en cada "batch" de datos. En consecuencia, el número de iteraciones en cada "epoch" dependerá del número de "batches". Durante cada iteración, el modelo hace predicciones sobre el "batch" de datos, calcula la pérdida (la diferencia entre las predicciones del modelo y las etiquetas reales), y luego ajusta los parámetros del modelo para minimizar esta pérdida.

Con la realización de este experimento pretendíamos analizar como influía el número de "epochs" en los diferentes modelos de "unlearning". Para llevar a cabo el experimento, decidimos elegir las edades que se encontraban entre 20 y 28 años (como en el experimento descrito en la Sección 4.4) y utilizar los modelos de "unlearning" descritos en la Sección 3.2 para eliminar los datos del modelo original.

4.5.1. Resultados Utility

Con el fin de comparar los resultados obtenidos con los distintos modelos, decidimos evaluar en primer lugar las predicciones en el conjunto de prueba, usando la "utility".

Los resultados que esperamos obtener son en primer lugar, que la precisión del modelo original $\mathcal{M}_O$ sea mayor a la del modelo entrenado desde cero $\mathcal{M}_R$ (ya que se entrena con más datos), por lo que el valor que obtengamos debería ser superior a 1, hecho que se refleja en la Tabla 4.

En segundo lugar, debería de estar el modelo entrenado con un "fine-tuning" básico $\mathcal{M}_{FT}$, ya que es el que menos ruido introduce en el modelo y por tanto el impacto en las predicciones no debería ser muy elevado. Por último, debería estar el modelo entrenado a través de un "fine-tuning" con etiquetas aleatorias $\mathcal{M}_{RND}$, ya que es el que más cambia las representaciones internas del modelo.

Para evaluar los modelos calculamos el valor de "Utility", tal y como se describe en la Sección 4.3.1. A continuación se muestran los valores obtenidos para el modelo original o $\mathcal{M}_O$, el modelo de "fine-tuning básico" o $\mathcal{M}_{FT}$ y "fine-tuning" con etiquetas aleatorias o $\mathcal{M}_{RND}$. El modelo entrenado desde cero no se muestra ya que es el que se utiliza de referencia para las comparaciones.

Tabla 4: Resultados de Utility para diferentes epochs, evaluado en el conjunto de prueba.

Nº Epochs	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$
2	-	1,25	0,75
4	-	1,24	0,41
8	-	1,21	0,43
16	-	1,18	0,45
142	**1.28**	-	-

Como se puede apreciar en la Tabla 4, los resultados siguen el comportamiento esperado. El modelo con mejor rendimiento es el original, con un valor por encima de 1.

A continuación está el modelo de "fine-tuning" básico, donde el rendimiento disminuye respecto al original. Se puede apreciar que a medida que incrementa el numero de "epochs", es decir, se realiza "unlearning" por más tiempo, el rendimiento disminuye. Este fenómeno se debe a que el modelo va ajustando progresivamente su configuración, lo que implica un impacto en sus predicciones.

Finalmente encontramos el modelo de "fine-tuning" con etiquetas modificadas aleatoriamente, el cual introduce una gran distorsión en las predicciones incluso con pocas "epochs". A medida que aumenta el número de "epochs", también disminuye la utilidad, hasta cierto punto en el que aprende nuevas características asociando las imágenes con las etiquetas aleatorias que hemos introducido. Aunque estas características pueden mejorar levemente la utilidad, el rendimiento general sigue estando muy por debajo de los otros modelos.

En la Tabla 5 se detalla el MAE dado por los modelos en el conjunto de prueba, utilizados para calcular los valores mostrados en la Tabla 4.

Tabla 5: MAE promedio para diferentes epochs y modelos en el conjunto de prueba.

Nº Epochs	$\mathcal{M}_O$	$\mathcal{M}_R$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$
2	-	-	4,93	8,2
4	-	-	4,96	15,09
8	-	-	5,09	14,3
16	-	-	5,21	13,59
142	**4,81**	-	-	-
102	-	6,17	-	-

4.5.2. Resultados Forgetting

Una vez evaluado el rendimiento de los modelos en el conjunto de prueba, es esencial analizar su eficacia para eliminar la información no deseada del modelo. Para este propósito, emplearemos la métrica de "Forgetting" detallada en la Sección 4.3.2.

Los resultados esperados son que el modelo de "fine-tuning" básico ocupe la mejor posición, ya que aunque no logre eliminar completamente la influencia de los datos, gradualmente reduce su impacto. En segundo lugar, se espera que se sitúe el modelo original, seguido por el modelo de "fine-tuning" con etiquetas aleatorias, cuyas predicciones deben alejarse considerablemente de las etiquetas reales debido a la alta distorsión introducida. A continuación se muestran los resultados obtenidos.

Tabla 6: Resultados de Forgetting para diferentes epochs.

Nº Epochs	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$
2	-	0,75	0,65
4	-	0,75	0,249
8	-	0,78	0,242
16	-	**0,79**	0,246
142	0,672	-	-

En la Tabla 6 se muestran los valores de "Forgetting" obtenidos por los diferentes modelos. La revisión de los datos revela un comportamiento consistente con nuestras expectativas en cuanto al rendimiento de los modelos. Es decir, el modelo de "fine-tuning" básico se sitúa en primer lugar, ya que logra eliminar la información de manera gradual, sin impactar significativamente en las predicciones. Se puede apreciar que el mejor resultado se obtiene con 16 "epochs", evidenciando que a mayor proceso de "unlearning", mejor capacidad para olvidar datos.

En segundo lugar se encuentra el modelo original, que no es superado por el "fine-tuning" con etiquetas aleatorias, independientemente del número de "epochs" utilizado por el último. Incluso con 2 "epochs", este introduce una alta distorsión,

alejándose las predicciones obtenidas por el modelo de referencia (entrenado desde cero o $\mathcal{M}_R$).

En las Figuras 11 y 12 mostramos unos gráficos que ilustran el MAE en el conjunto de test $\mathcal{D}_t$ de los diferentes modelos cogiendo 100 predicciones al azar fijadas para todos los modelos:

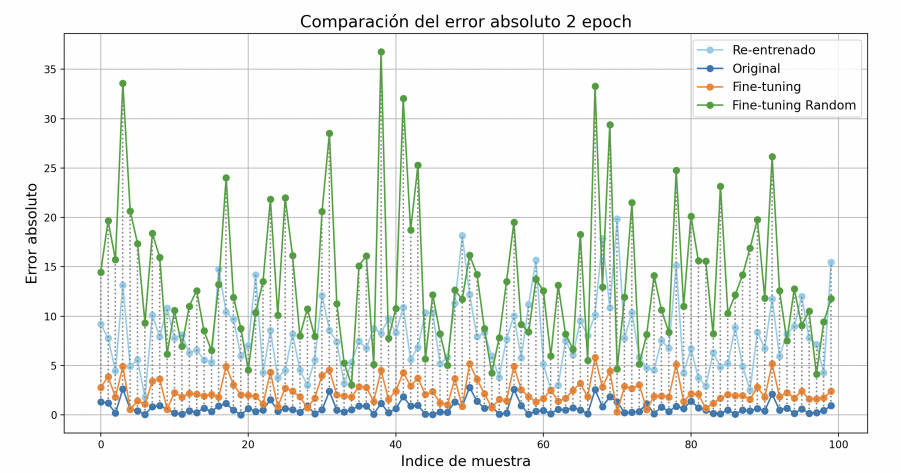

Figura 11: MAE para modelos de unlearning entrenados durante 2 epochs.

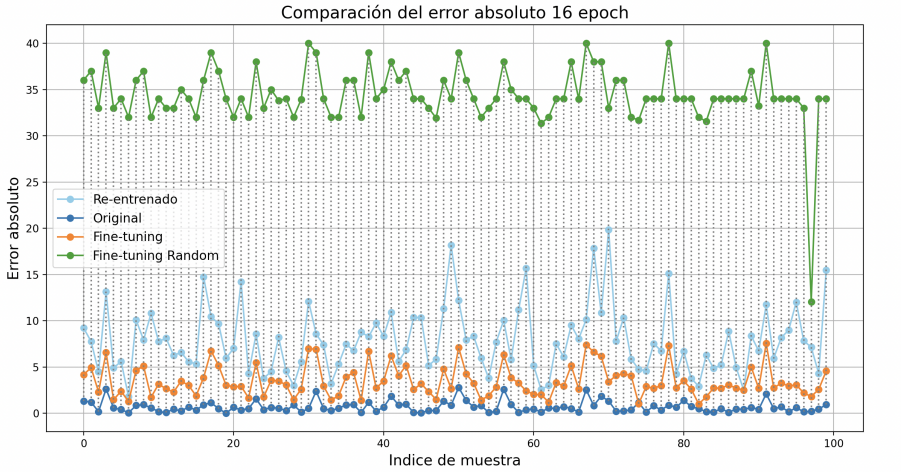

Figura 12: MAE para modelos de unlearning entrenados durante 16 epochs.

Tal y como se muestra en los gráficos (Figura 11 y 12), los resultados de los modelos de "unlearning" evaluados se comportan de manera diferente, al aumentar el numero de "epochs". Es decir, el modelo de "fine-tuning" básico $\mathcal{M}_{FT}$ obtiene un MAE que se sitúa entre el modelo original o $\mathcal{M}_O$ y el modelo entrenado desde cero o $\mathcal{M}_R$, tanto para 2 como para 16 "epochs", mostrando cierta consistencia. Se evidencia por tanto que la técnica de "fine-tuning" básico continua olvidando gradualmente los datos. Por otro lado, el modelo de "fine-tuning" con etiquetas aleatorias $\mathcal{M}_{RND}$ introduce perturbaciones que afectan considerablemente las predicciones. Se puede observar en los gráficos como aumenta el MAE con el paso de

las “epochs”. Este comportamiento da a entender que el modelo ha perdido toda capacidad de generalizar correctamente y hacer predicciones que se ajusten con los datos reales.

4.5.3. Resultados combinando Utility y Forgetting

Al haber evaluado los resultados según las métricas de Utility y Forgetting, podemos combinar ambas para obtener el modelo con mejor rendimiento al olvidar datos pero que mantenga cierta precisión.

Tabla 7: Valores de F = Utility * Forgetting (Ecuación 4.5) obtenidos para diferentes modelos y epochs.

Epochs	UTILITY			FORGETTING			F		
-	ORIG.	FT	RND	ORIG.	FT	RND	ORIG.	FT	RND
2	-	1,25	0,75	-	0,75	0,65	-	0,93	0,49
4	-	1,24	0,41	-	0,75	0,249	-	0,94	0,10
8	-	1,21	0,43	-	0,78	0,242	-	**0,95**	0,10
16	-	1,18	0,45	-	0,79	0,246	-	0,94	0,11
142	1,28	-	-	0,67	-	-	0,86	-	-

La Tabla 7 muestra los resultados obtenidos al combinar ambas métricas tal y como se explica en la Sección 4.3.3. Una vez finalizado el análisis de las tablas, podemos concluir que el mejor modelo de “unlearning” es el de “fine-tuning” básico durante 8 “epochs”. Este consigue mantener un equilibrio entre precisión en el conjunto de datos que no se deben olvidar y capacidad para eliminar información.

En cuanto al método de “fine-tuning” con etiquetas aleatorias, el mejor modelo es el obtenido al entrenar durante 2 “epochs”, ya que es el que menos perturbaciones introduce en el modelo y por tanto sus predicciones se mantienen mas fieles a las de los valores reales. Cabe destacar que la puntuación obtenida por esta última técnica esta por debajo a la del modelo original, lo que evidencia su bajo rendimiento.

4.6. Experimento 3: cambiando el conjunto de forget set

Como se menciona en la Sección 4.4, dividimos el conjunto original de datos en cinco grupos diferentes, cada uno abarcando un rango de ocho años. Dado que las muestras para cada año presentaban una distribución no balanceada, aplicamos la técnica de “data augmentation” descrita en la Sección 4.2.3. Este procedimiento permite al modelo mejorar los resultados al entrenar con más muestras en los grupos con menos datos. Sin embargo, es probable que la precisión en las edades que originalmente contaban con un mayor número de datos sea más alta que en las demás. Por este motivo, decidimos llevar a cabo este experimento. Nos permitirá analizar el comportamiento de los modelos de “unlearning” en los diferentes grupos de edades y evaluar cómo afecta la distribución de los datos a la precisión de las predicciones.

Para llevar a cabo el experimento, decidimos utilizar los modelos en los que mejores resultados obtuvimos en la Sección 4.4, que son el "fine-tuning" ($\mathcal{M}_{FT}$) entrenado durante 8 "epochs" y el "fine-tuning" con etiquetas aleatorias ($\mathcal{M}_{RND}$) entrenado durante 2 "epochs". La Tabla 8 muestra los resultados obtenidos en los modelos según las métricas de "Utility", "Forgetting" y "F" definidas en la Sección 4.3.

El modelo original $\mathcal{M}_O$, se entrenó con el conjunto de entrenamiento original $\mathcal{D}_o$, y únicamente los modelos de "unlearning" se entrenaron utilizando los diferentes conjuntos de "retain set" $\mathcal{D}_r^j$.

Tabla 8: Utility, Forgetting y F obtenidos al entrenar diferentes modelos con diferentes conjuntos de datos.

Forget set	Retain Set	Utility			Forgetting			F		
-	-	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$	$\mathcal{M}_O$	$\mathcal{M}_{FT}$	$\mathcal{M}_{RND}$
0 [20-28]	[29-60]	1,28	1,21	0,75	0,67	0,78	0,65	0,86	**0,94**	0,49
1 [28-36]	$[20, 28) \cup (36, 60]$	1,02	1,04	0,79	0,79	0,80	0,78	0,81	**0,83**	0,61
2 [36-44]	$[20, 36) \cup (44, 60]$	1,01	1,03	0,95	0,79	0,81	0,76	0,80	**0,83**	0,72
3 [44-52]	$[20, 44) \cup (52, 60]$	1,04	1,05	0,95	0,73	0,74	0,70	0,76	**0,77**	0,67
4 [52-60]	$[20 - 51]$	1,14	1,13	1,16	0,53	0,59	0,54	0,60	**0,67**	0,63

Una vez analizados los datos de la Tabla 8, observamos que el modelo con mejor rendimiento es $\mathcal{M}_{FT}$, es decir, el "fine-tuning" básico, que se mantiene estable para todos los conjuntos de "forget/retain" evaluados. Este resultado se debe a que, al ir olvidando los datos de manera progresiva, no introduce perturbaciones significativas que alteren el modelo en exceso, logrando así buenos resultados.

En segundo lugar, se encuentra el modelo $\mathcal{M}_O$, el modelo original. Por último, el modelo con peor rendimiento es $\mathcal{M}_{RND}$, el fine-tuning con etiquetas aleatorias. Esto se debe a que el modelo $\mathcal{M}_{RND}$ introduce una gran perturbación en el modelo, alterando significativamente su configuración interna y por tanto, sus predicciones. Esta perturbación no solo falla en su objetivo de ayudar al modelo a olvidar los datos, sino que también provoca una pérdida considerable en la capacidad de generalización, resultando en un rendimiento inferior al del modelo original, donde no se ha realizado ningún tipo de "unlearning".

El único caso en el que no se describe el comportamiento mencionado anteriormente es en el grupo de edades de 52 a 60 años, donde el modelo $\mathcal{M}_{RND}$ supera el rendimiento del modelo $\mathcal{M}_O$. Analizando los datos de toda la fila, observamos que $\mathcal{M}_{RND}$ tiene el "Utility" más alto para ese grupo de edad, mientras en el "Forgetting" sigue obteniendo la peor puntuación. Posiblemente esto se debe a que las perturbaciones introducidas por el modelo ayudan a mejorar la generalización para ese caso.

Un dato interesante es que se observa una notable caída en los valores de "Forgetting" al eliminar las muestras entre 52 y 60 años del conjunto de entrenamiento original. Aunque no disponemos de evidencias concretas, deducimos que este comportamiento se debe a que esta categoría es atípica. Es decir, las imágenes de personas entre 52 y 60 años varían más que las de personas entre 20 y 44 años. Por esta

razón, al eliminar la categoría de 52 a 60 años, el modelo resultante presenta mayores diferencias con respecto a un modelo entrenado desde cero sin estos datos, en comparación con los otros grupos de edad. Esta mayor variabilidad en las imágenes de personas mayores hace que la eliminación de sus datos sea una tarea mas difícil para el modelo de "unlearning", o que se requiera mas iteraciones o "epochs".

Para concluir esta sección, gracias a los resultados obtenidos se puede afirmar que a pesar de existir otros métodos más efectivos para el "unlearning", un método basado en el "fine-tuning" puede ser una buena base para estudiar el desaprendizaje automático.

5. Conclusiones y trabajos futuros

En este trabajo, revisamos algunos de los métodos existentes de desaprendizaje automático, una línea de investigación emergente en el ámbito del aprendizaje automático. Evaluamos dos métodos sencillos y de bajo consumo de recursos computacionales en un problema de reconocimiento de edad. Para valorar los resultados, definimos una métrica sencilla y eficaz, que ofrece un buen rendimiento sin requerir tiempos de cálculo prolongados ni la necesidad de entrenar múltiples modelos repetidamente.

Gracias a haber conseguido completar los objetivos iniciales que nos propusimos, y haber podido realizar los experimentos detallados anteriormente, podemos concluir que el modelo de "unlearning" que utiliza la técnica de "fine-tuning" básico o $\mathcal{M}_{FT}$ es el que obtiene un mejor rendimiento a lo largo de los diferentes experimentos, basándonos en las métricas definidas en el apartado 4.3 y los modelos evaluados. Este resultado se debe a que el proceso de eliminación de información se lleva a cabo de manera progresiva y muy lenta, lo que minimiza el impacto negativo en la precisión de las predicciones. Sin embargo, la desventaja de este enfoque es que existe la posibilidad de que nunca consiga borrar los datos en su totalidad.

Por otro lado, el modelo de "unlearning" con "fine-tuning" aleatorio o $\mathcal{M}_{RND}$, resulta ser totalmente ineficiente al cambiar las etiquetas aleatoriamente, al menos de la forma en la que lo implementamos. Como se observa en los experimentos, generalmente muestra resultados con un rendimiento inferior al del modelo original $\mathcal{M}_O$, al cual no se le ha aplicado ninguna técnica de desaprendizaje. Este hecho sugiere que se introducen perturbaciones tan significativas que provocan un desajuste en la configuración interna del modelo, resultando en una pérdida considerable de su capacidad para predecir resultados con eficacia.

Según el análisis realizado, podemos afirmar que además de obtener resultados coherentes en nuestro estudio, existe un largo camino por recorrer hasta conseguir implementar modelos de "unlearning" totalmente efectivos y eficientes, ya que la línea de investigación aún está en una fase inicial.

5.1. Limitaciones del trabajo

Como se menciona en la Sección 3, la limitación de recursos computacionales ha tenido un papel determinante en el momento de escoger que técnicas de "unlearning" utilizar. En la Sección 2 se plantean diversas posibilidades a la hora de aplicar el proceso de desaprendizaje. Sin embargo, algunas de las técnicas presentadas requieren una elevada cantidad de recursos en cuanto a GPU's.

Para implementar nuestros modelos, hemos utilizado la herramienta de google Colab, que dispone de manera gratuita de una GPU con una capacidad limitada de 15 GB. El punto negativo, es que esta se bloquea al superar este límite, de la misma manera que el entorno también se bloquea al pasar un determinado tiempo de ejecución. Al bloquearse el entorno, se debe esperar como mínimo un período de 12 horas, lo que provoca que entrenar un modelo lleve una gran cantidad de

tiempo. Por ese motivo, decidimos implementar los modelos de "fine-tuning" que menos recursos consumían.

5.2. Trabajos futuros

Los posibles siguientes pasos para este trabajo consistirían en implementar modelos de "unlearning" mas potentes, así como considerar otras aplicaciones aparte del reconocimiento de edad, utilizando para ello un entorno de ejecución apto. De esta manera se tendría un conocimiento mucho más amplio sobre el comportamiento de las diferentes técnicas y se podría elaborar un análisis mas detallado para establecer con mayor seguridad cuál es la técnica más efectiva.

Otro posible enfoque al cual dirigir este trabajo, es el área de "Explainable Artificial Intelligence" [36]. Actualmente el "machine learning" se centra en desarrollar algoritmos que permitan aprender y hacer predicciones basadas en datos. A diferencia de enfoques tradicionales de programación, en los que se codifican explícitamente todas las reglas y lógica, los sistemas de "machine learning" se entrenan utilizando grandes conjuntos de muestras. El objetivo es que la máquina identifique patrones y relaciones en los datos. Esta técnica se basa en una "black box" o caja negra ya que la toma de decisiones no es transparente ni fácil de interpretar para los humanos. "Explainable Artificial Intelligence" contrasta con este enfoque y se centra en crear modelos de IA que sean más transparentes y comprensibles, con la finalidad que sean fácilmente interpretables para las personas. Utilizando dicha técnica en nuestro trabajo se podría intentar encontrar la manera en que la máquina crea relaciones entre las diferentes edades, y posteriormente utilizar esta información para facilitar el olvido de los datos, o ayudar a comprender mejor cómo funciona el proceso de desaprendizaje.

Referencias

[1] S. Mercuri, R. Khraishi, R. Okhrati, D. Batra, C. Hamill, T. Ghasempour, and A. Nowlan, "An introduction to machine unlearning," *ArXiv*, vol. abs/2209.00939, 2022. [Online]. Available: https://arxiv.org/abs/2209.00939

[2] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," *CoRR*, vol. abs/2209.02299, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2209.02299

[3] A. Cohen, A. D. Smith, M. Swanberg, and P. N. Vasudevan, "Control, confidentiality, and the right to be forgotten," *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[4] S. Alelyani, "Detection and evaluation of machine learning bias," *Applied Sciences*, 2021.

[5] N. G. Marchant, B. I. P. Rubinstein, and S. Alfeld, "Hard to forget: Poisoning attacks on certified machine unlearning," *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, vol. 36, no. 7, pp. 7691–7700, 2022.

[6] E. Triantafillou, F. Pedregosa, J. Hayes, P. Kairouz, I. Guyon, M. Kurmanji, G. K. Dziugaite, P. Triantafillou, K. Zhao, L. S. Hosoya, V. D. Julio C. S. Jacques Junior, I. Mitliagkas, S. Escalera, J. Wan, S. Dane, M. Demkin, and W. Reade, "NeurIPS 2023 - Machine Unlearning," 2023. [Online]. Available: https://kaggle.com/competitions/neurips-2023-machine-unlearning

[7] A. Sekhari, J. Acharya, G. Kamath, and A. Suresh, "Remember what you want to forget: Algorithms for machine unlearning," *ArXiv*, 2021. [Online]. Available: https://arxiv.org/abs/2103.03279

[8] J. Xu, Z. Wu, C. Wang, and X. Jia, "Machine unlearning: Solutions and challenges," *IEEE Transactions on Emerging Topics in Computational Intelligence*, p. 1–19, 2024. [Online]. Available: http://dx.doi.org/10.1109/TETCI.2024.3379240

[9] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. S. Kankanhalli, "Deep regression unlearning," *IEE*, vol. abs/2210.08196, 2022.

[10] L. Bourtoule, V. Chandrasekaran, C. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *Proceedings - 2021 IEEE Symposium on Security and Privacy, SP 2021*, ser. Proceedings - IEEE Symposium on Security and Privacy. United States: Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 141–159.

[11] K. Koch and M. Soll, "No matter how you slice it: Machine unlearning with sisa comes at the expense of minority classes," *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 622–637, 2023.

[12] J. P. Lalor, H. Wu, and H. Yu, "Improving machine learning ability with fine-tuning," *ArXiv*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:7256103

[13] S. H. S. Basha, S. K. Vinakota, V. Pulabaigari, S. Mukherjee, and S. Dubey, "Autotune: Automatically tuning convolutional neural networks for improved transfer learning," *Neural networks : the official journal of the International Neural Network Society*, vol. 133, pp. 112–122, 2020.

[14] S. Marullo, M. Tiezzi, M. Gori, S. Melacci, and T. Tuytelaars, "Continual learning with pretrained backbones by tuning in the input space," *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2023.

[15] A. Warnecke, L. Pirch, C. Wressnegger, and K. Rieck, "Machine unlearning of features and labels," in *Network and Distributed System Security (NDSS) Symposium 2023: 27 February - 3 March 2023, San Diego, CA, USA*, 2023.

[16] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Fast yet effective machine unlearning," *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–10, 2024. [Online]. Available: http://dx.doi.org/10.1109/TNNLS.2023.3266233

[17] G. Battistelli, D. Selvi, and A. Tesi, "Robust switching control: Stability analysis and application to active disturbance attenuation," *IEEE Transactions on Automatic Control*, vol. 62, pp. 6369–6376, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:20728840

[18] Y. Liu, C. Sun, Y. Wu, and A. Zhou, "Unlearning with fisher masking," *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2310.05331

[19] H. Sajjad, N. Durrani, and F. Dalvi, "Neuron-level Interpretation of Deep NLP Models: A Survey," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 1285–1303, 11 2022. [Online]. Available: https://doi.org/10.1162/tacl_a_00519

[20] H. C. Moon, S. Joty, and X. Chi, "Gradmask: Gradient-guided token masking for textual adversarial example detection," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 3603–3613. [Online]. Available: https://doi.org/10.1145/3534678.3539206

[21] C. Fan, J. Liu, Y. Zhang, E. Wong, D. Wei, and S. Liu, "Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2310.12508

[22] F. Pérez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," *Department of Electrical Engineering, Princeton University*, 2008.

[23] X. Li, D. Chang, T. Tian, and J. Cao, "Large-margin regularized softmax cross-entropy loss," *IEEE Access*, vol. 7, pp. 19 572–19 578, 2019.

[24] A. Agarwala, J. Pennington, Y. Dauphin, and S. Schoenholz, "Temperature check: theory and practice for training models with softmax-cross-entropy losses," *ArXiv*, 2020. [Online]. Available: https://arxiv.org/abs/2010.07344

[25] H. Xu, T. Zhu, L. Zhang, W. Zhou, and P. S. Yu, "Machine unlearning: A survey," *ACM Comput. Surv.*, vol. 56, no. 1, aug 2023.

[26] A. K. Tarun, V. S. Chundawat, M. Mandal, and M. Kankanhalli, "Deep Regression Unlearning," *arXiv*, Oct. 2022. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2022arXiv221008196T

[27] I. Bratko, "Machine learning: between accuracy and interpretability," in *Learning, Networks and Statistics*, ser. CISM Courses and Lectures, G. D. Riccia, H.-J. Lenz, and R. Kruse, Eds. Wien; York: Springer, 1997, vol. 382, pp. 163–177, [COBISS.SI-ID 1171028].

[28] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2016.

[29] Preeti and I. Khan, "Membership inference attacks on machine learning model," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2022.

[30] H. Hu, Z. Salcic, L. Sun, G. Dobbie, P. Yu, and X. Zhang, "Membership inference attacks on machine learning: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, pp. 1 – 37, 2021.

[31] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, and L. Zettlemoyer, "Detecting pretraining data from large language models," *arXiv*, 2024. [Online]. Available: https://arxiv.org/abs/2310.16789

[32] S. Goel, A. Prabhu, and P. Kumaraguru, "Evaluating inexact unlearning requires revisiting forgetting," *ArXiv*, 2022. [Online]. Available: https://arxiv.org/abs/2201.06640

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[34] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[35] M. Danelljan, T. B. Schön, F. K. Gustafsson, and G. Bhat, "Dctd: Deep conditional target densities for accurate regression," *arXiv*, 2024. [Online]. Available: http://arxiv.org/abs/1909.12297

[36] S. A. and S. R., "A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends," *Decision Analytics Journal*, vol. 7, p. 100230, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S277266222300070X