

# Reidentificación de personas invariante a la edad mediante redes profundas y análisis de su eficiencia frente a redes de una capa

Sergio Quijano Rey  
Javier Merí de la Maza <sup>1</sup> Pablo Mesejo Santiago <sup>2</sup>

Universidad de Granada <sup>1</sup>Departamento de Análisis Matemático  
<sup>2</sup>Departamento de Ciencias de la Computación e Inteligencia Artificial

28 de Junio del 2024



UNIVERSIDAD  
DE GRANADA

# Contenidos

## 1 Estudio sobre la expresividad de las redes neuronales

- Objetivos
- Tarea de aprendizaje
- Modelización de las redes no profundas
- Modelización de las redes profundas
- Resultados principales

## 2 Reconocimiento facial invariante a la edad

- Objetivos
- Tarea a resolver
- Enfoque
- Experimentación preliminar
- Solución original
- Validación experimental de nuestra solución
- Conclusiones

## 3 Bibliografía



# Contenidos

## 1 Estudio sobre la expresividad de las redes neuronales

- Objetivos
- Tarea de aprendizaje
- Modelización de las redes no profundas
- Modelización de las redes profundas
- Resultados principales

## 2 Reconocimiento facial invariante a la edad

- Objetivos
- Tarea a resolver
- Enfoque
- Experimentación preliminar
- Solución original
- Validación experimental de nuestra solución
- Conclusiones

## 3 Bibliografía

## Objetivos

- Estudiar la mayor potencia expresiva de las redes profundas frente a las redes no profundas.
  - Modelizar una **tarea de aprendizaje automático**.
  - Modelizar **dos tipos de arquitecturas** de aprendizaje automático a partir de ciertas descomposiciones tensoriales.
    - Arquitectura **no profunda** o somera, utilizando la descomposición *CP*.
    - Arquitectura **profunda**, empleando la descomposición *HT*.
  - Dar dos resultados para estudiar el fenómeno de **eficiencia en profundidad** y cómo de frecuentemente ocurre.



## Tarea de aprendizaje

- Buscamos resolver una tarea de **clasificación de imágenes**.
  - Representamos las imágenes de entrada como **parches**,  $(\vec{x}_1, \dots, \vec{x}_N)$  con  $\vec{x}_i \in \mathbb{R}^S$ . Esta representación se utiliza en la práctica [DBK<sup>+</sup>20].
  - Clasificamos la imagen de entrada como el valor  $y$  para el cual se maximiza la **función de puntuación**  $h_y : \mathbb{R}^S \times \dots \mathbb{R}^S \rightarrow \mathbb{R}$ .
  - Por lo tanto, buscamos aprender  $Y$  funciones de puntuación a partir de los datos y las arquitecturas de aprendizaje automático que desarrollemos.

## Función de puntuación

- **Funciones de representación**  $\{f_d(\vec{x}) : d \in \mathbb{N}\} \subseteq L^2(\mathbb{R}^S)$ .  
El conjunto de funciones será total y linealmente independiente.
    - Neuronas.
    - *Radial Basis Functions* (Gaussianas).
  - Expresamos las combinaciones lineales finitas como:

$$h_y(\vec{x_1}, \dots, \vec{x_N}) \approx \sum_{d_1, \dots, d_N \in \mathbb{N}} \mathcal{A}_{d_1, \dots, d_N}^y \prod_{i=1}^N f_{d_i}(\vec{x_i}). \quad (1)$$

- En [CSS15] se justifica empíricamente que al trabajar con imágenes podemos tomar  $M = 100$  con lo que se verifica

$$h_y(\vec{x_1}, \dots, \vec{x_N}) = \sum_{\substack{d_1, \dots, d_N=1}}^M \mathcal{A}_{d_1, \dots, d_N}^y \prod_{i=1}^N f_{\theta_{d_i}}(\vec{x_i}). \quad (2)$$



# Descomposición CANDECOMP/PARAFAC

- Aplicar la descomposición *CP* en el tensor de coeficientes de (2).

## Descomposición CANDECOMP/PARAFAC

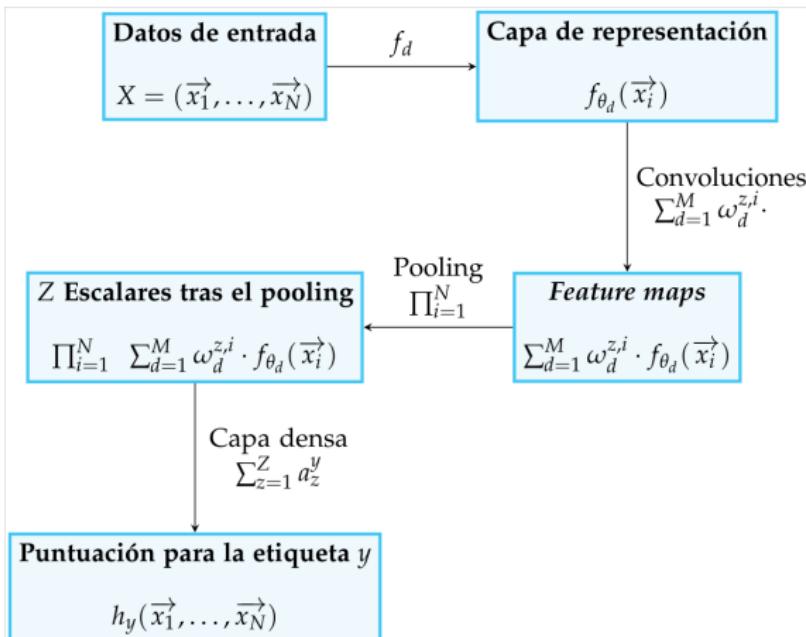
Todo tensor  $\mathcal{A}$  puede ser expresado como la suma de tensores puros. Es decir,  $\forall \mathcal{A} \in \mathbb{R}^{M_1 \times \dots \times M_N}, \exists Z \in \mathbb{N}$ :

$$\mathcal{A} = \sum_{i=1}^Z \overrightarrow{v_i^{(1)}} \otimes \dots \otimes \overrightarrow{v_i^{(N)}}; \quad \overrightarrow{v_i^{(k)}} \in \mathbb{R}^{M_k}, \quad \forall i \in \Delta_Z, \quad \forall k \in \Delta_N. \quad (3)$$

- Rango *CP*.
- Nuestro modelo queda como:

$$h_y(\vec{x_1}, \dots, \vec{x_N}) = \sum_{z=1}^Z a_z^y \prod_{i=1}^N \sum_{d=1}^M \omega_d^{z,i} f_{\theta_d}(\vec{x_i}). \quad (4)$$

## Relación con arquitecturas de aprendizaje automático



**Figura:** Representación gráfica del modelo *CP*.



# Descomposición *Hierarchical Tucker*

$$\phi^{1,j,\gamma} := \sum_{\alpha=1}^{r_0} a_\alpha^{1,j,\gamma} \cdot \overrightarrow{\varphi^{2j-1,\alpha}} \otimes \overrightarrow{\varphi^{2j,\alpha}}$$

...

$$\phi^{l,j,\gamma} := \sum_{\alpha=1}^{r_{l-1}} a_\alpha^{l,j,\gamma} \cdot \phi^{l-1,2j-1,\alpha} \otimes \phi^{l-1,2j,\alpha} \quad (5)$$

...

$$\mathcal{A}^y := \sum_{\alpha=1}^{r_{L-1}} a_\alpha^{L,y} \cdot \phi^{L-1,1,\alpha} \otimes \phi^{L-1,2,\alpha}$$

- $l$ : nivel en la descomposición.
- $j$ : posición dentro del nivel  $l$ .
- $\gamma$ : tensor de la capa  $l$  y posición  $j$ .
- $r_l$ : cuántos tensores hay en cada posición  $j$  de la capa  $l$ .



# Primer ejemplo

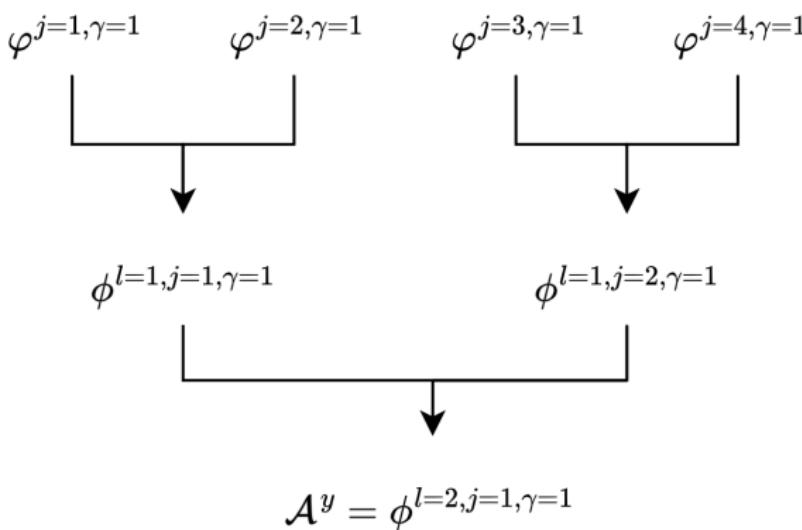


Figura: Ejemplo gráfico tomando  $r = 1$ ,  $L = 2$ .



## Segundo ejemplo

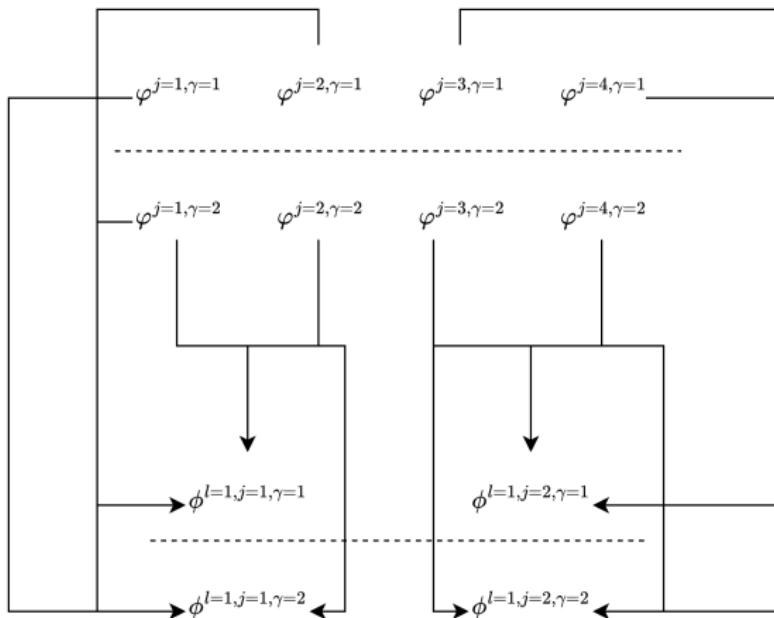


Figura: Ejemplo gráfico tomando  $r = 2$ ,  $L = 2$  capas, primer paso.



## Segundo ejemplo

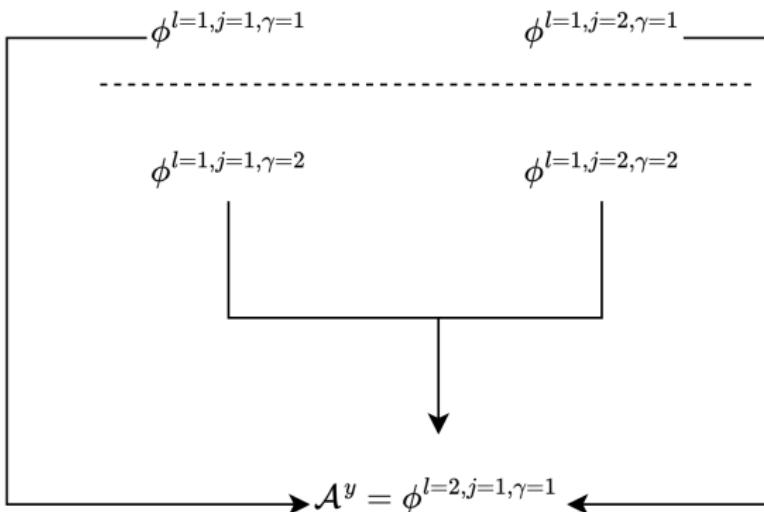


Figura: Ejemplo gráfico tomando  $r = 2$ ,  $L = 2$  capas, segundo paso.



# Relación con arquitecturas de aprendizaje automático

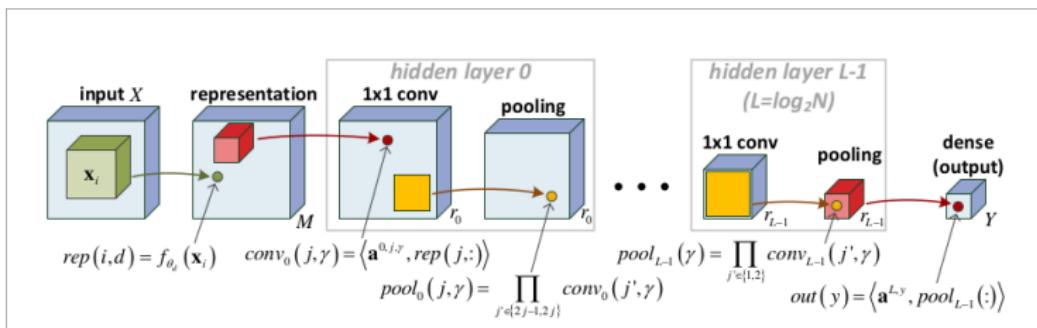


Figura: Funcionamiento de nuestro modelo. Imagen extraída de [CSS15].



# Eficiencia en profundidad

- La **eficiencia en profundidad** es el fenómeno en el que una arquitectura polinomial obtiene una potencia expresiva que en otro modelo requiere un número exponencial de parámetros.

## Penalización en número de parámetros

Dado un tensor  $\mathcal{A}^y$  como resultado de un modelo  $CP$ , realizar dicho tensor con un modelo  $HT$  requiere de:

$$N \cdot Z^2 \cdot \frac{2^{L-1} - 1}{2^{L-1}} \xrightarrow{L \rightarrow \infty} N \cdot Z^2. \quad (6)$$

coeficientes adicionales.

- ¿Y al revés?



# Primer resultado central

## Rango *CP* exponencial de un modelo *HT*

Sea  $\mathcal{A}^y \in \mathcal{T}_{N,M}$  dado por las ecuaciones (5). Definamos  $r := \min\{r_0, M\}$  y consideremos el espacio de todas las posibles configuraciones de parámetros de nuestro modelo *HT*  $\{\overrightarrow{a^{I,j,\gamma}}\}_{I,j,\gamma}$ . En este espacio, el tensor generado  $\mathcal{A}^y$  tendrá rango *CP* de al menos  $r^{N/2}$  casi por doquier. Es decir, el conjunto de parámetros del modelo *HT* con los que el modelo tiene rango *CP* menor que  $r^{N/2}$  tiene medida nula. El resultado se mantiene si forzamos los coeficientes compartidos en la ecuación (5). Es decir, haciendo  $\overrightarrow{a^{I,\gamma}} \equiv \overrightarrow{a^{I,j,\gamma}}$  y considerando el espacio de configuraciones  $\{\overrightarrow{a^{I,\gamma}}\}_{I,\gamma}$ .



## Segundo resultado central

Incapacidad del modelo *CP* para aproximar eficientemente el modelo *HT*

Dado un conjunto de funciones de representación linealmente independientes,  $\{f_{\theta_d} : d \in \Delta_M\}$ , aleatorizar los pesos de un modelo *HT* (5) a partir de una distribución de probabilidad continua induce funciones de puntuación  $h_y$  que con probabilidad uno no pueden ser aproximadas arbitrariamente bien (en el sentido  $L^2$ ) por un modelo *CP* con menos de  $r := \min\{r_0, M\}^{N/2}$  canales. Este resultado se mantiene forzando los coeficientes compartidos en el modelo *HT* mientras que dejamos el modelo *CP* sin restricciones.



- El primer resultado nos dice que casi todos los tensores  $\mathcal{A}^y$  que podemos generar con un modelo  $HT$  tienen rango  $CP$  de al menos  $r^{N/2}$ , lo que implica que el modelo  $CP$  necesita un número exponencial de parámetros.
- El segundo resultado añade a este hecho que ni siquiera pueden aproximarse eficientemente (menos de un número exponencial de coeficientes) por una descomposición  $CP$ .
- Damos información precisa sobre cómo de frecuente ocurre este hecho (casi por doquier). Otros trabajos [HK09] únicamente dan ejemplos concretos en los que esto ocurre.



# Contenidos

## 1 Estudio sobre la expresividad de las redes neuronales

- Objetivos
- Tarea de aprendizaje
- Modelización de las redes no profundas
- Modelización de las redes profundas
- Resultados principales

## 2 Reconocimiento facial invariante a la edad

- Objetivos
- Tarea a resolver
- Enfoque
- Experimentación preliminar
- Solución original
- Validación experimental de nuestra solución
- Conclusiones

## 3 Bibliografía

# Objetivos

- Resolver una tarea de **reconocimiento facial invariante a cambios en la edad**, por sus siglas en inglés, *AIFR*.
- Estudiar dos **variantes online** en la función de pérdida *Triplet Loss*.



AIFR



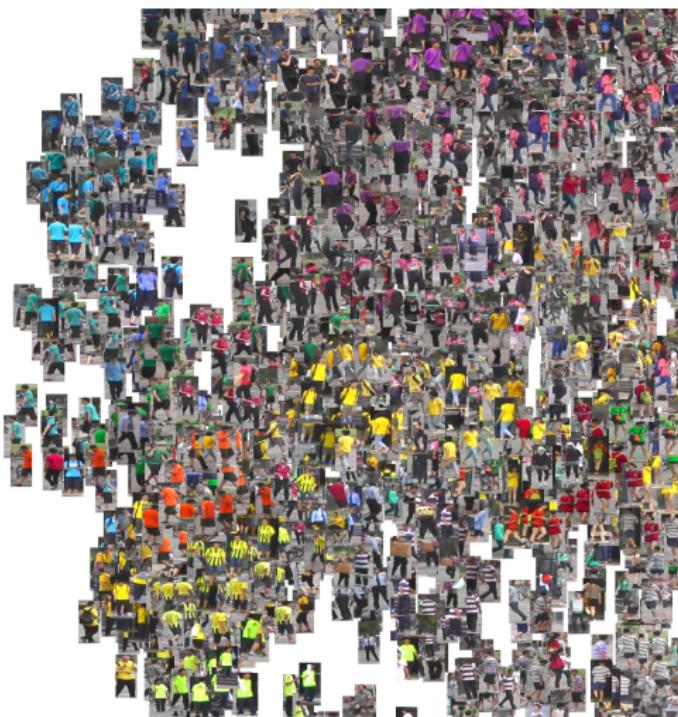
**Figura:** Ejemplo de datos con los que trabajamos en una tarea de *AIFR*. Imagen extraída de [SB18].

# Problemas asociados a la tarea

- Pueden ser más similares dos personas distintas de la misma edad que la misma persona en dos edades muy distantes.
- El envejecimiento modifica las características faciales.
- Trabajar con identidades nunca vistas.
- Escasez de conjuntos de datos para estudiar la tarea de *AIFR*, que además presentan diversas dificultades.



# Embedding semántico



- Queremos que nuestra red aprenda un **embedding semántico**.
- Elementos de la misma identidad deberán estar cerca entre sí, mientras que elementos de distintas identidades deberán estar distantes.



# Redes siamesas

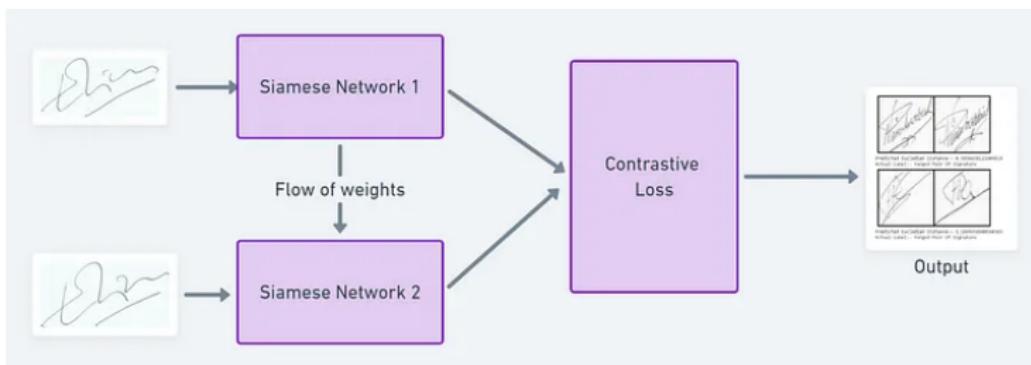


Figura: Imagen extraída de [Ben].

- Herramienta para aprender el *embedding*.
- Se utiliza o *contrastive loss* (pares) o *triplet loss* (triples).



# Triplet Loss

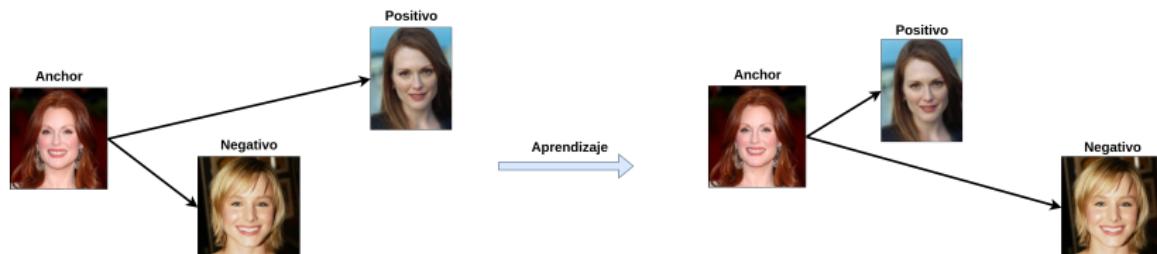


Figura: Imágenes extraídas de [inf14].

A partir de:

$$D_{A,P} \leq D_{A,N}, \quad (7)$$

llegamos a:

$$\mathcal{L}_{tri}(\theta; A, P, N) := \max\{D_{A,P} - D_{A,N} + \alpha, 0\} \quad (8)$$



# Variantes *online* sobre *Triplet Loss*

- Problema: necesitamos generar los triples de forma *offline*.
- Solución: generar los *batches* de forma *online*.
  - Usando *P-K sampling*.
  - Aplicando las variantes *Batch All* y *Batch Hard* sobre estos nuevos *batches*

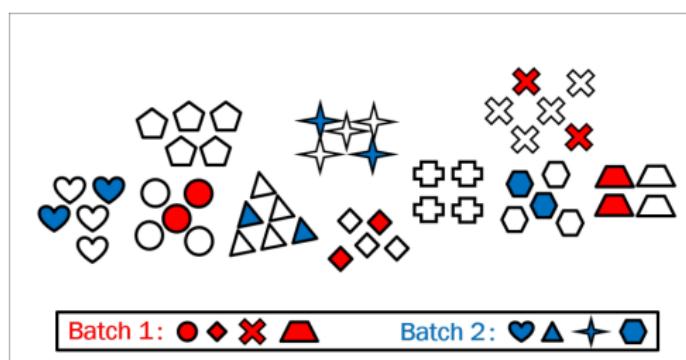


Figura: Imagen extraída de [LS21].



## Variantes *online*

Sobre el anterior *P-K* sampling:

- **Batch All:** probar todas las combinaciones ancla - positivo - negativo.
- **Batch Hard:** por cada ancla, computar la pérdida con el positivo más lejano y el negativo más cercano (combinación más complicada).



# CACD

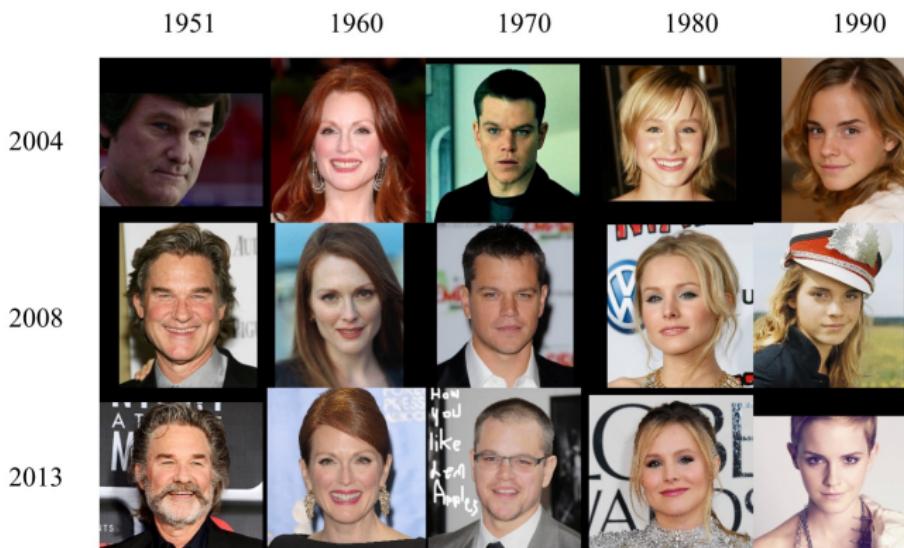


Figura: Imagen extraída de [CCH14]



# Resultados preliminares sobre CACD

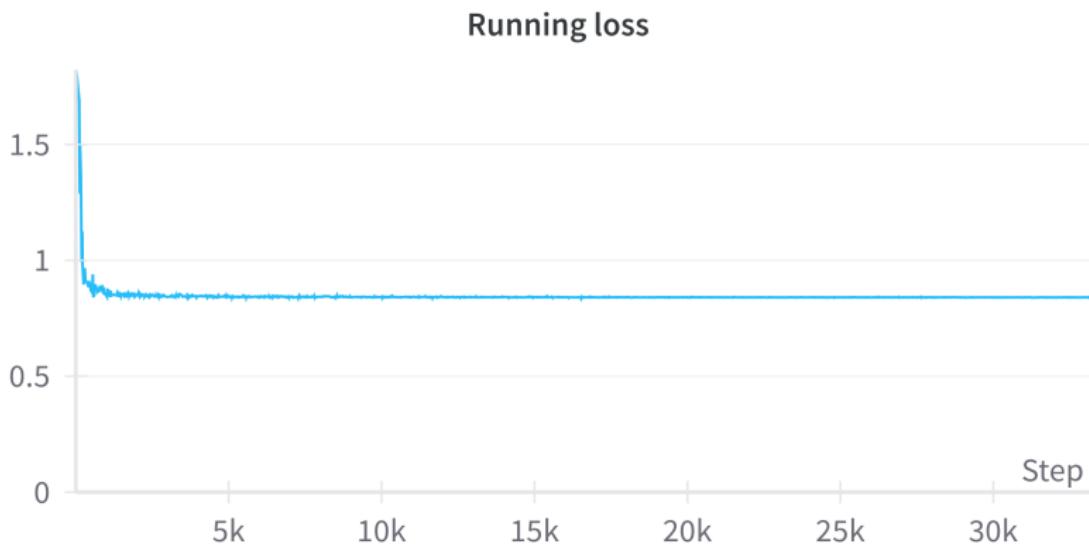


Figura: Error durante el entrenamiento



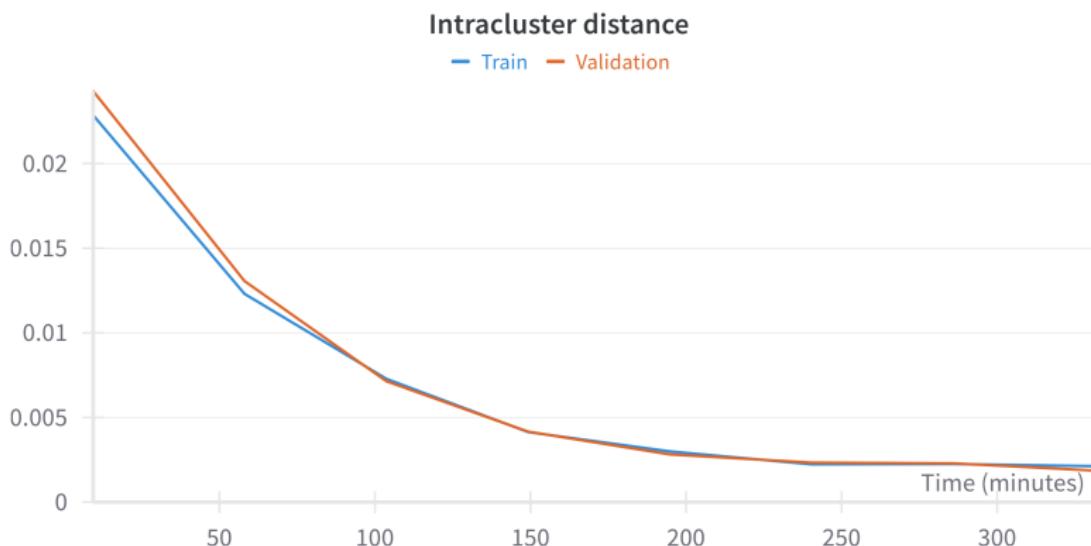


Figura: Distancia *intracluster* media



# Resultados preliminares sobre CACD

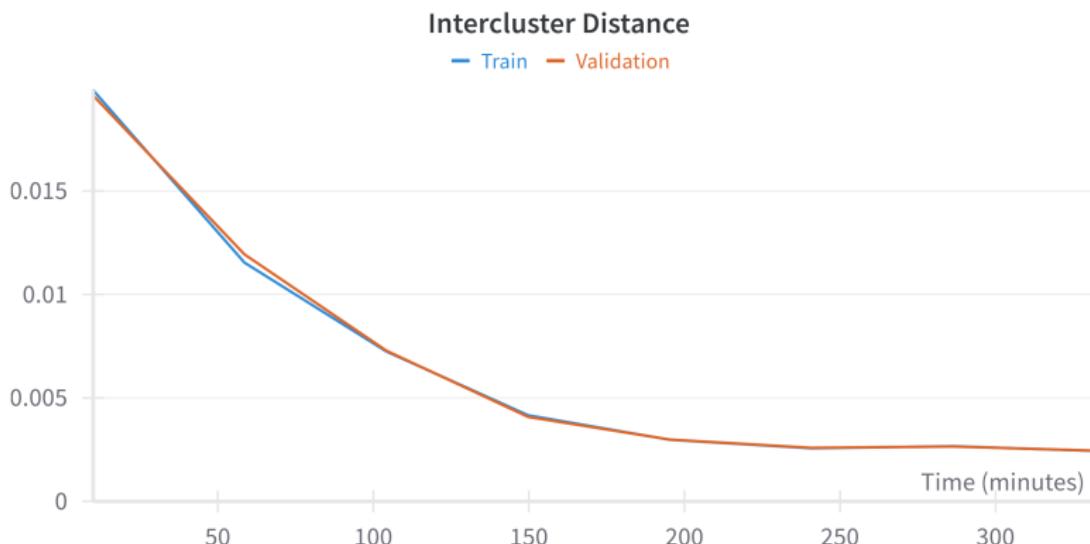


Figura: Distancia *intercluster* media



# Resultados preliminares sobre CACD

Conjunto	Rank@1 Accuracy	Rank@5 Accuracy	Silhouette
Entrenamiento	0.07185	0.15244	-0.41858
Test	0.01000	0.00100	-0.32984

**Cuadro:** Valores de distintas métricas obtenidas sobre el conjunto CACD de entrenamiento y sobre el conjunto FG-Net de test.



# Resultados preliminares sobre *MNIST*

Conjunto	<i>Rank@1 Accuracy</i>	<i>Rank@5 Accuracy</i>	<i>Silhouette</i>
Entrenamiento	0.0937	0.453	0.000
Test	0.085	0.414	0.000

**Cuadro:** Métricas de evaluación obtenidas tras entrenar el modelo sobre *MNIST*.

En la experimentación realizada por [Bie] observamos los mismos problemas que estamos exponiendo.



# Raíz del problema y nuestra propuesta de solución

- La red aprende a transformar todas las entradas a un vector no nulo del *embedding*.
- En esta situación tenemos que:

$$\begin{aligned}\mathcal{L}(a, p, n) &= \text{ReLU}(d(a, p) - d(a, n) + \alpha) \\ &= \text{ReLU}(d(\vec{v}_0, \vec{v}_0) - d(\vec{v}_0, \vec{v}_0) + \alpha) \\ &= \text{ReLU}(0 - 0 + \alpha) = \text{ReLU}(\alpha) = \alpha.\end{aligned}\tag{9}$$

- Penalizar las distancias negativas pequeñas dividiendo por su media en el término ancla - positivo.

$$\mathcal{L}(a, p, n) = \widehat{\text{ReLU}}((\widehat{d}(a, p) - \widehat{d}(a, n)) / \text{mean}(d(a, n)) + \alpha)\tag{10}$$



# Resultados en *MNIST* con nuestra solución



Figura: Función de pérdida.



# Resultados en *MNIST* con nuestra solución

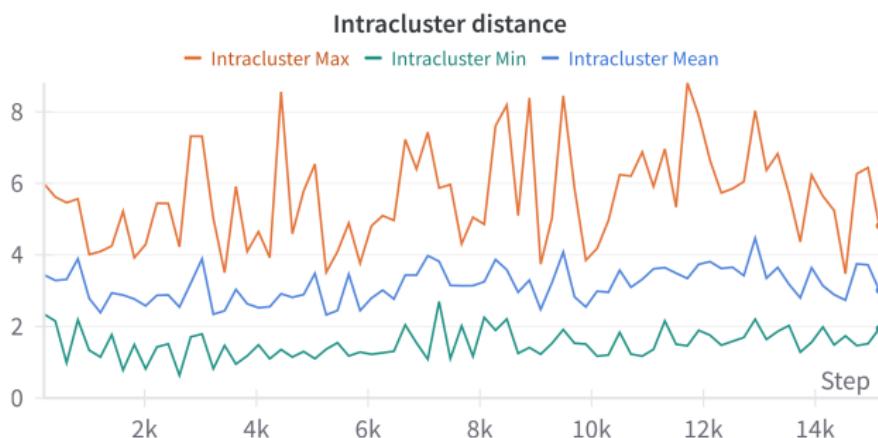


Figura: Distancias intracluster.



# Resultados en *MNIST* con nuestra solución

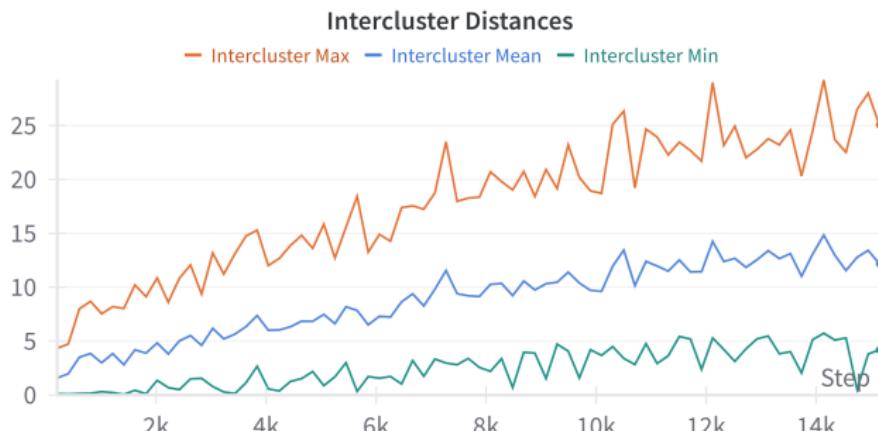


Figura: Distancias intercluster.



# Mejora de los resultados en *MNIST*

Métrica	Conjunto	Antes	Después	Mejora
<i>Rank@1 Accuracy</i>	Entrenamiento	0.0937	0.997	10.64
<i>Rank@1 Accuracy</i>	Test	0.085	0.992	11.67
<i>Rank@5 Accuracy</i>	Entrenamiento	0.453	0.997	2.20
<i>Rank@5 Accuracy</i>	Test	0.414	0.991	2.39
<i>Silhouette</i>	Entrenamiento	0.000	0.953	$\infty$
<i>Silhouette</i>	Test	0.000	0.922	$\infty$



# Mejora de los resultados en CACD

Métrica	Conjunto	Antes	Después	Mejora
<i>Rank@1 Accuracy</i>	Entrenamiento	0.0092	0.2676	29.09
<i>Rank@1 Accuracy</i>	Test	0.0312	0.3732	11.96
<i>Rank@5 Accuracy</i>	Entrenamiento	0.0136	0.4639	34.11
<i>Rank@5 Accuracy</i>	Test	0.0781	0.6152	7.88
<i>Silhouette</i>	Entrenamiento	-0.1366	-0.1648	-0.02
<i>Silhouette</i>	Test	-0.1596	-0.1832	-0.023



# Conclusiones

- Hemos identificado un problema de diseño en las variantes *online* de *Triplet Loss*.
- Hemos propuesto una solución original y validado experimentalmente su gran eficacia.
- Obtenemos buenos resultados en la tarea de *AIFR*.



# Contenidos

## 1 Estudio sobre la expresividad de las redes neuronales

- Objetivos
- Tarea de aprendizaje
- Modelización de las redes no profundas
- Modelización de las redes profundas
- Resultados principales

## 2 Reconocimiento facial invariante a la edad

- Objetivos
- Tarea a resolver
- Enfoque
- Experimentación preliminar
- Solución original
- Validación experimental de nuestra solución
- Conclusiones

## 3 Bibliografía



[Ben]

Sean Benhurst.

A friendly Introduction to Siamese Networks —  
towardsdatascience.com.

[https://towardsdatascience.com/](https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab1)

[a-friendly-introduction-to-siamese-networks-85ab1](https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab1)

Fecha de último acceso: 17-06-2024].

[Bie]

Adam Bielski.

Github - adambielski/siamese-triplet: Siamese and triplet networks with online pair/triplet mining in pytorch — [github.com](https://github.com/adambielski/siamese-triplet).

[https:](https://github.com/adambielski/siamese-triplet)

[//github.com/adambielski/siamese-triplet.](https://github.com/adambielski/siamese-triplet)

Fecha de último acceso: 01-06-2024.

[CCH14]

Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu.

Cross-age reference coding for age-invariant face  
recognition and retrieval.

2014.

- [CSS15] Nadav Cohen, Or Sharir, and Amnon Shashua.  
On the expressive power of deep learning: A tensor analysis.

2015.

- [DBK<sup>+</sup>20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby.  
An image is worth 16x16 words: Transformers for image recognition at scale.  
*arXiv.org*, 2020.

- [HK09] Wolfgang Hackbusch and Stefan Kühn.  
A new scheme for the tensor representation.  
*Journal of Fourier Analysis and Applications*,  
15(5):706–722, Oct 2009.



- [inf14] <https://bcsiriuschen.github.io/CARC/>, 2014.  
Fecha de último acceso: 16-09-2023.
- [LS21] [Shengcai Liao and Ling Shao](#).  
Graph sampling based deep metric learning for  
generalizable person re-identification, 2021.
- [SB18] [Manisha M. Sawant and Kishor M. Bhurchandi](#).  
Age invariant face recognition: a survey on facial aging  
databases, techniques and effect of aging.  
*Artificial Intelligence Review*, 52(2):981–1008, October  
2018.



# Reidentificación de personas invariante a la edad mediante redes profundas y análisis de su eficiencia frente a redes de una capa

Sergio Quijano Rey  
Javier Merí de la Maza <sup>1</sup> Pablo Mesejo Santiago <sup>2</sup>

Universidad de Granada <sup>1</sup>Departamento de Análisis Matemático  
<sup>2</sup>Departamento de Ciencias de la Computación e Inteligencia Artificial

28 de Junio del 2024



UNIVERSIDAD  
DE GRANADA