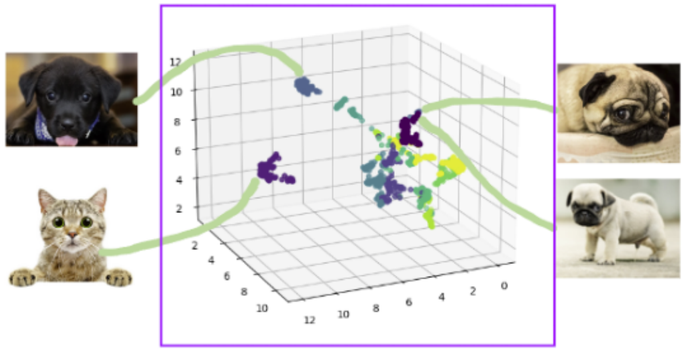
dcc CIENCIAS DE LA COMPUTACIÓN UNIVERSIDAD DE CHILE

Universidad de los Andes > FACULTAD DE INGENIERÍA Y CIENCIAS APLICADAS

Aprendizaje de Representaciones

[Representation Learning]



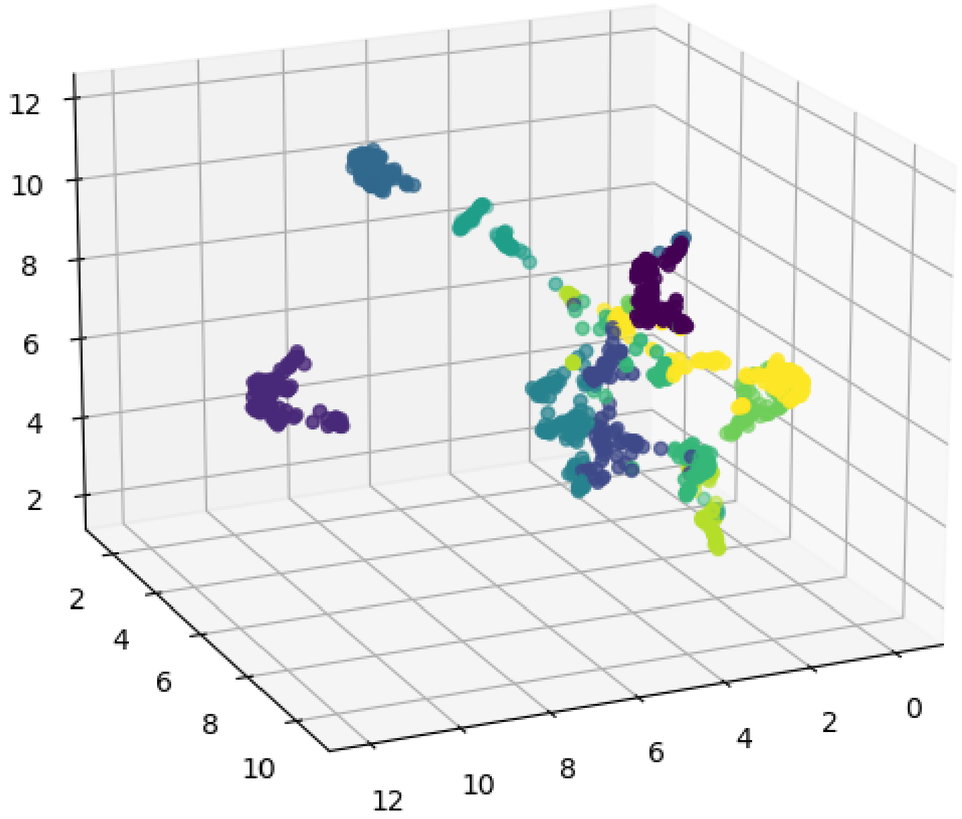
CLASE 4

José M. Saavedra R.
Profesor Asistente

jmsaavedrar@miuandes.cl

Ed. Ingeniería - Oficina 315

Representation Learning



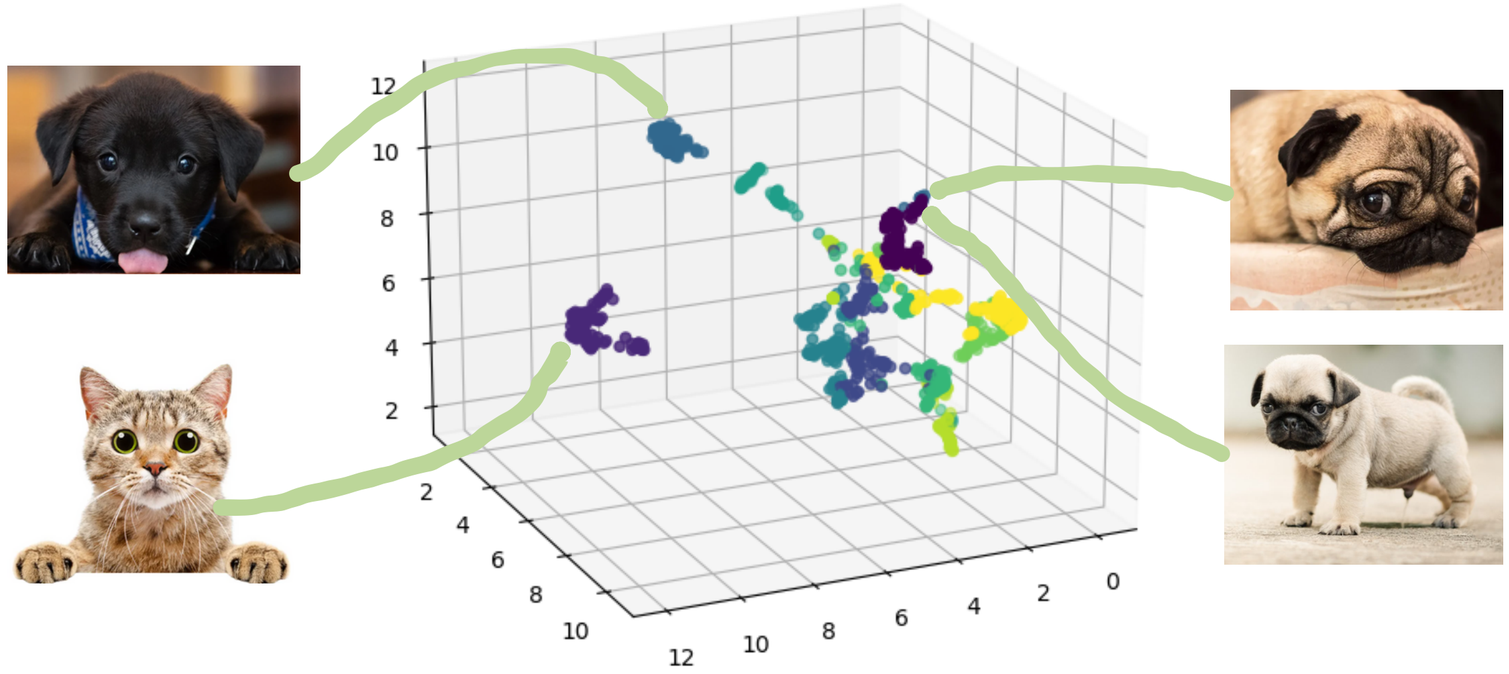
An effective and efficient representation (feature vector) is the **key**...

- Effective: representations lead to a semantic space.
- Efficient: low-dimensionality (as low as possible)

Representation Learning

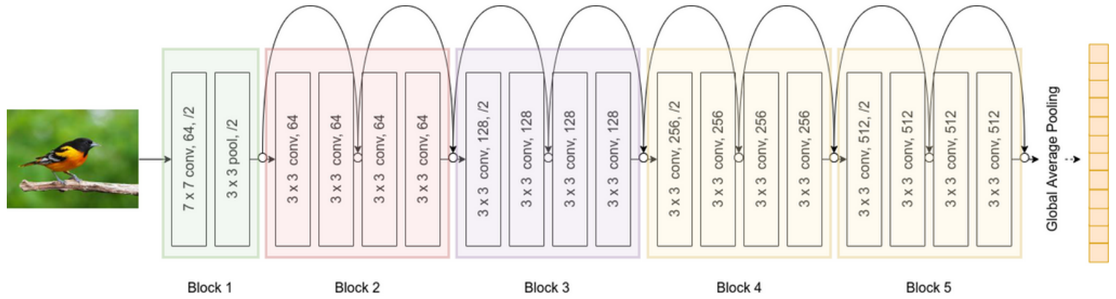
Effectiveness > a semantic space

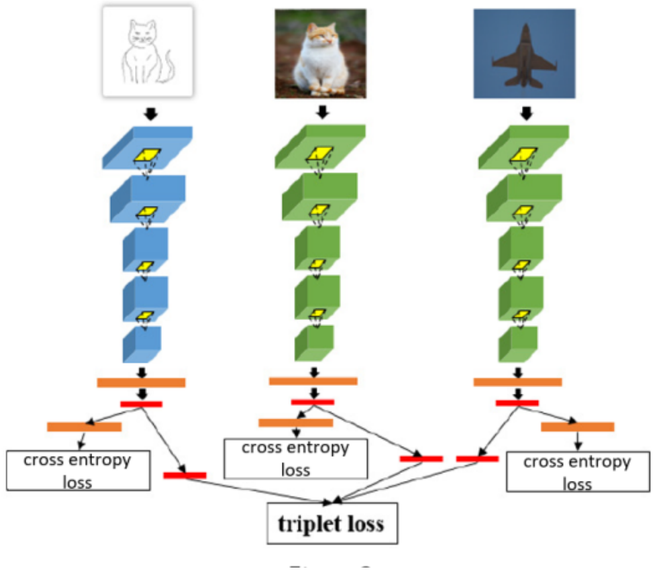
Efficiency > **a** low dimensionality

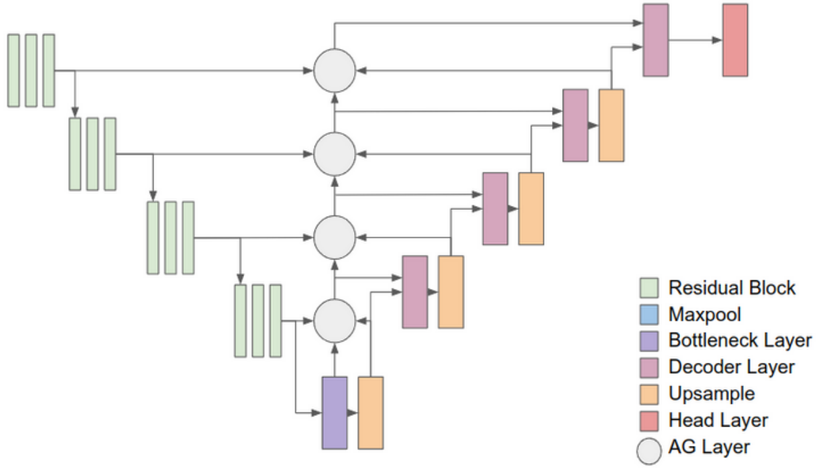


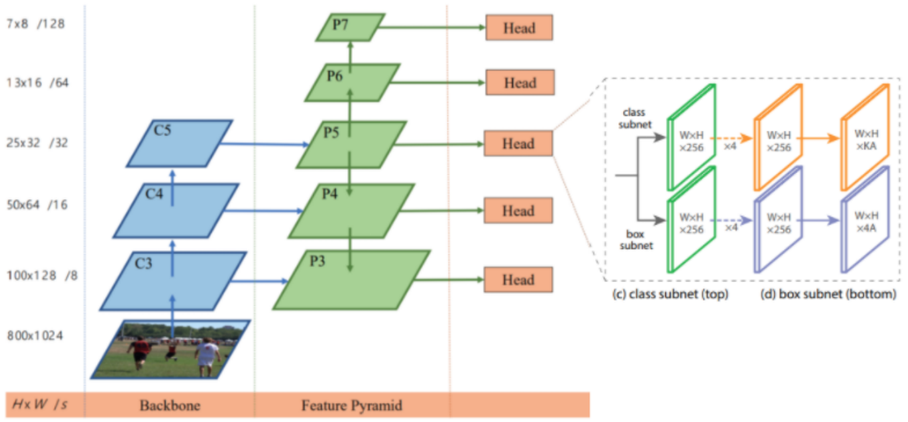
Representation Learning

The backbone







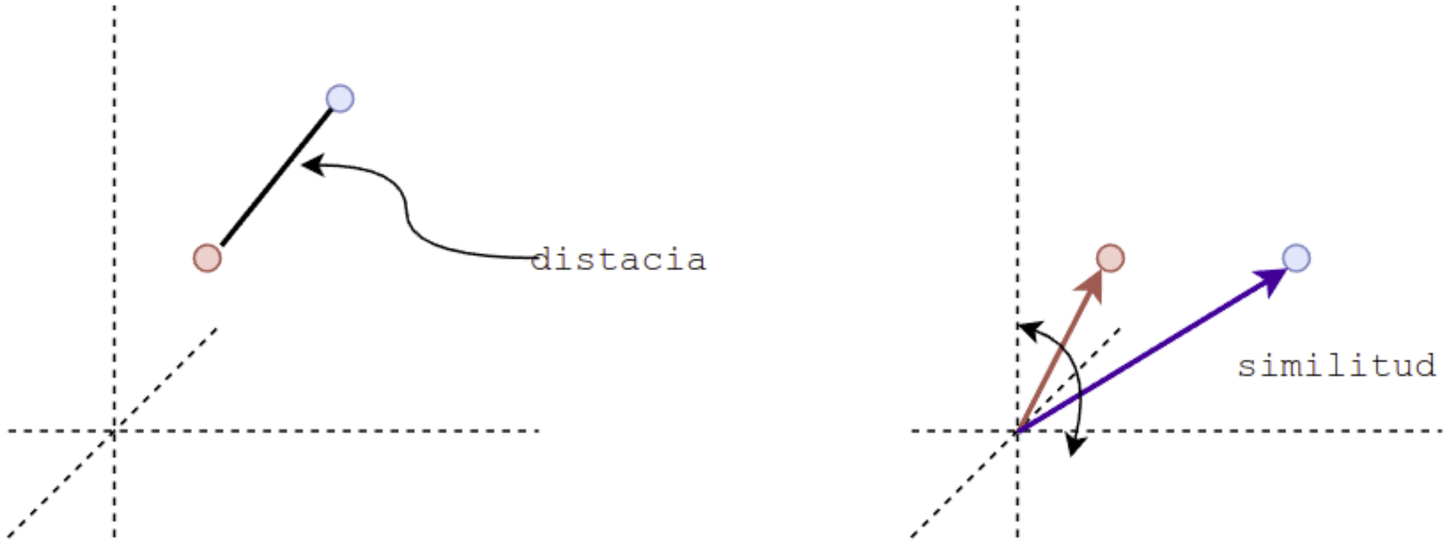


Representation Learning

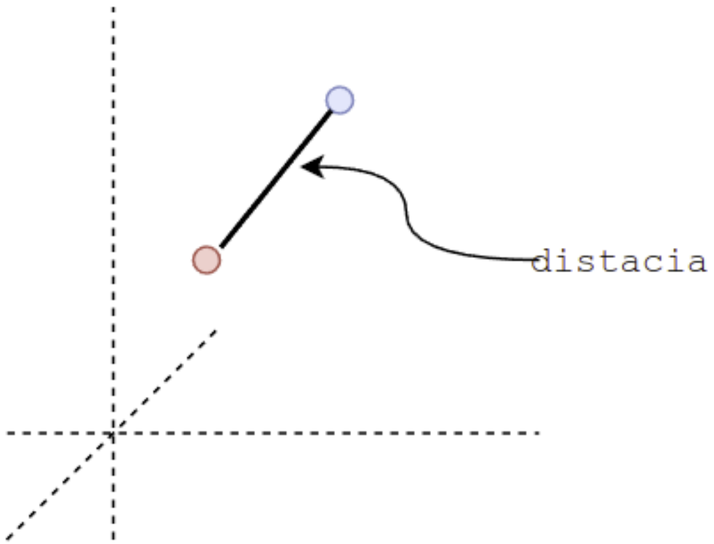
Comparación entre Vectores

Dado un espacio de características, es crítico definir una función que mida la cercanía entre los vectores (feature vectors). Para este fin, hay dos tipos de funciones:

- **Funciones de distancia** que miden qué tan lejos están dos puntos en el espacio
- **Funciones de similitud** que miden la semejanza entre vectores. En este último caso, la idea es medir la dirección del vector.



Representation Learning



Funciones de Distancia

Existe una familia de funciones de distancia muy utilizadas en este dominio, se trata de las distancias de **Minkowski**, definida bajo la siguiente expresión:

- Distancias de Minkowski:

$$L_p(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p\right)^{1/p}, p \geq 1$$

  - Manhattan (p=1)

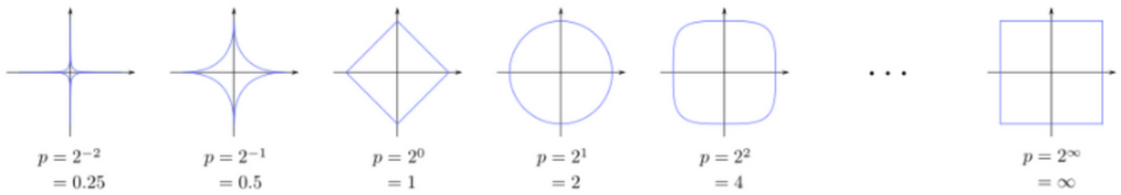
$$L_1(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

  - Euclidiana (p=2)

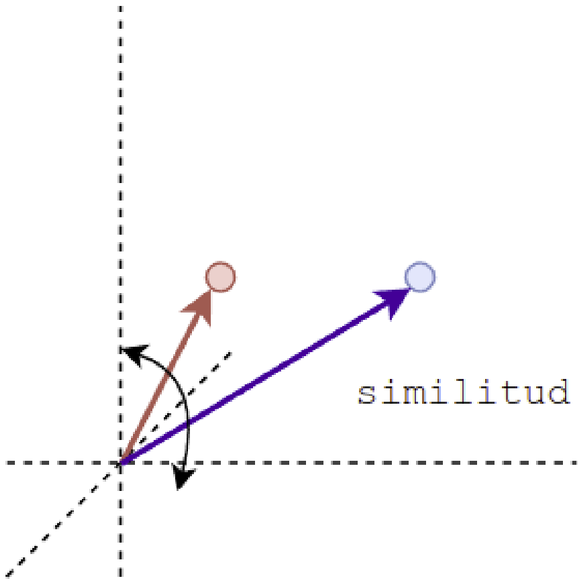
$$L_2(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^2\right)^{1/2}$$

  - Máximo (p=inf)

$$L_\infty(\mathbf{x},\mathbf{y}) = max_{i=1}^{d} |x_i - y_i|$$



Representation Learning



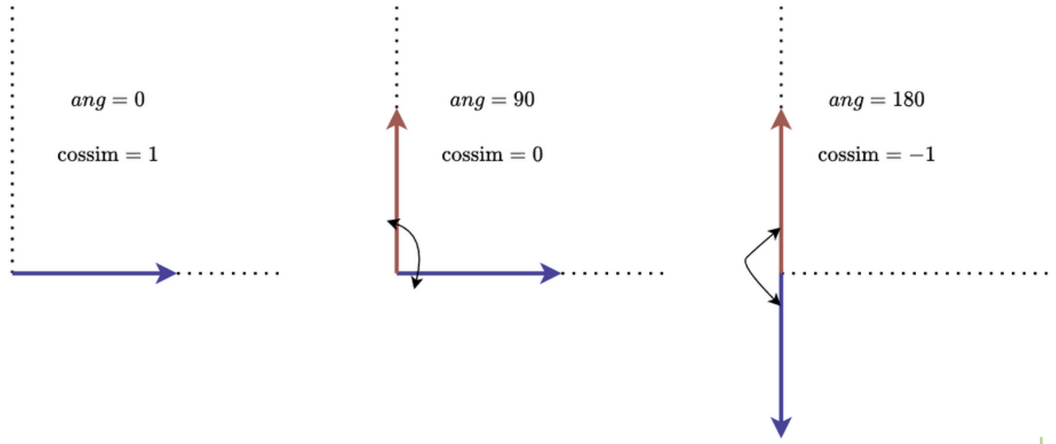
Funciones de Similitud

La función de similitud más usada es la similitud **coseno**. Este mide la similitud basada en la dirección de los vectores.

$$\text{cossim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||||\mathbf{v}||},$$

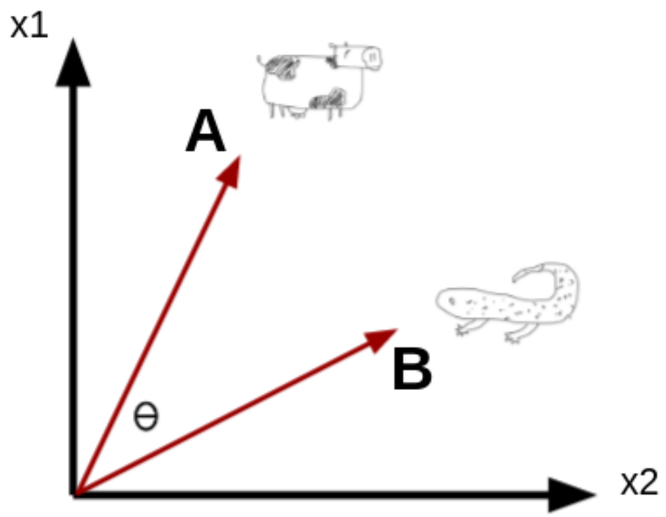
donde $||\cdot||$ define la norma o magnitud del vector. Esto es:

$$||\mathbf{u}|| = \sqrt{\sum_{i=1}^{d} u_i^2}$$



Representation Learning

Relación cossim - L2



$$sim(A, B) = cos(\theta)$$

$$cos(\theta) = \frac{A \cdot B}{||A||||B||}$$

Si suponemos que A y B son unitarios

||A|| = ||B|| = 1

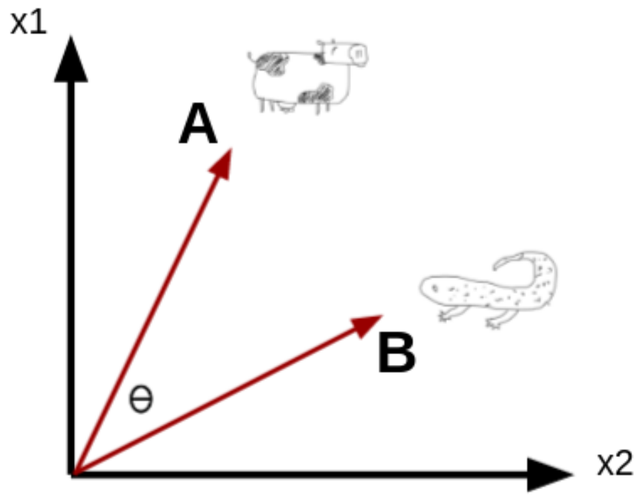
$$cos(\theta) = A \cdot B$$

$$sim(A, B) \in [-1, 1]$$

Representation Learning

Relación cossim - L2

Supondremos, A y B son unitarios ||A|| = ||B|| = 1



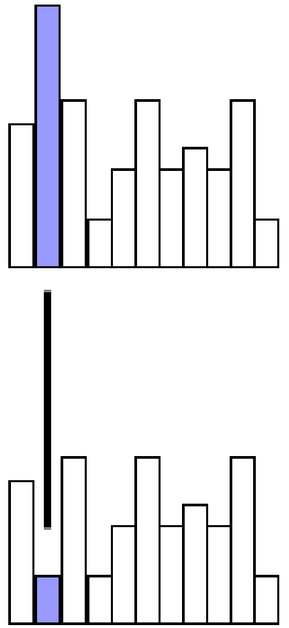
$$
\begin{aligned}
L_2(A,B) &= \sqrt{\sum_{i=1}^{d}(a_i - b_i)^2} \\
&= \sqrt{\sum_{i=1}^{d}(a_i^2 - 2a_ib_i + b_i^2)} \\
&= \sqrt{\sum_{i=1}^{d} a_i^2 - 2\sum_{i=1}^{d} a_ib_i + \sum_{i=1}^{d} b_i^2} \\
&= \sqrt{2 - 2\sum_{i=1}^{d} a_ib_i}
\end{aligned}
$$

$$L_2(A,B) = \sqrt{2 - 2cos(\theta)}$$

Representation Learning

Normalización

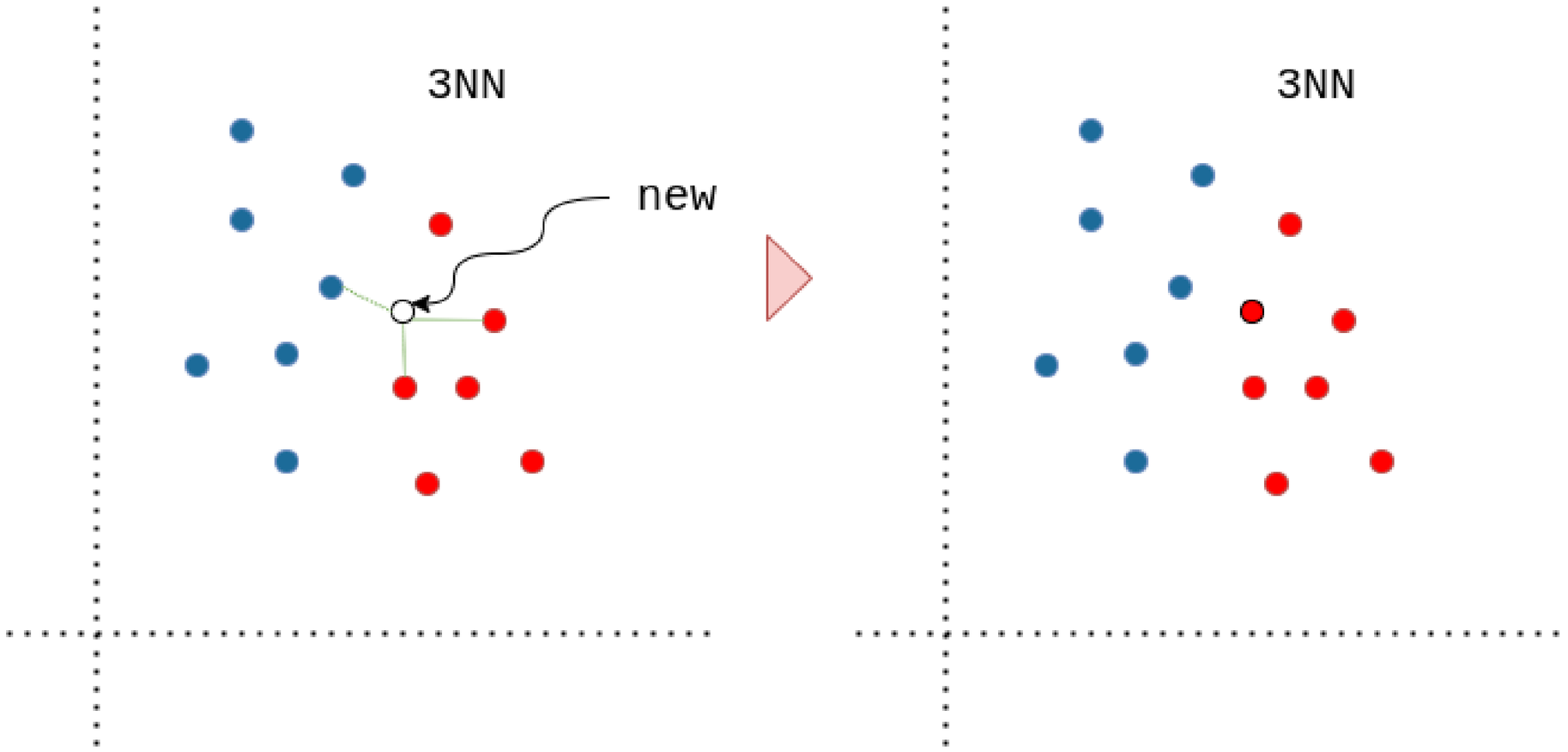
- **Bursting Effect**: Alta diferencia en pocas dimensiones genera alta diferencia entre los vectores.
- **Solución práctica:** Square-Root Normalization



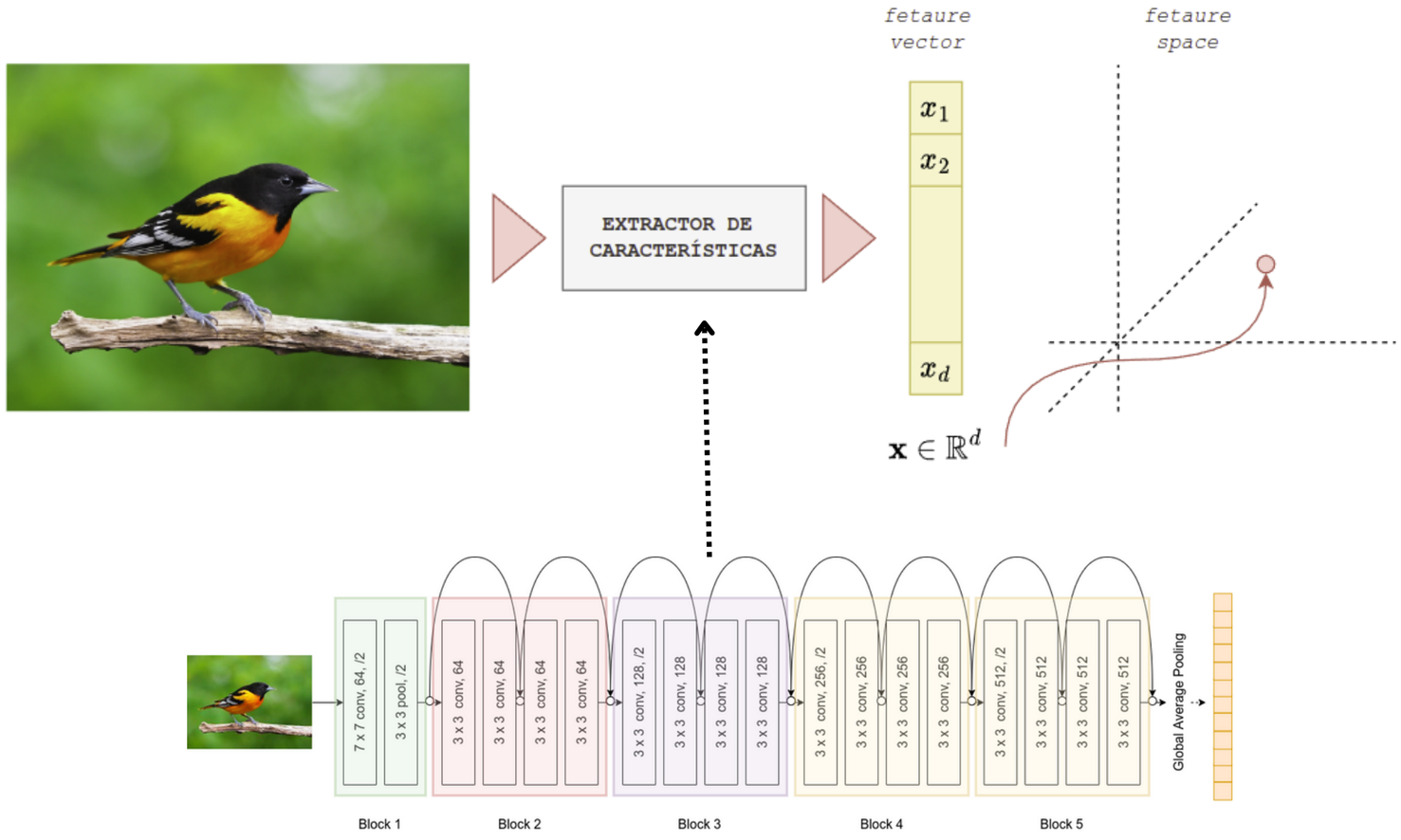
$$sr\_norm(\mathrm{x}) = unit(sign(\mathrm{x}) \cdot \sqrt{|\mathrm{x}|})$$

Representation Learning

k-Nearest Neighbors



Representation Learning

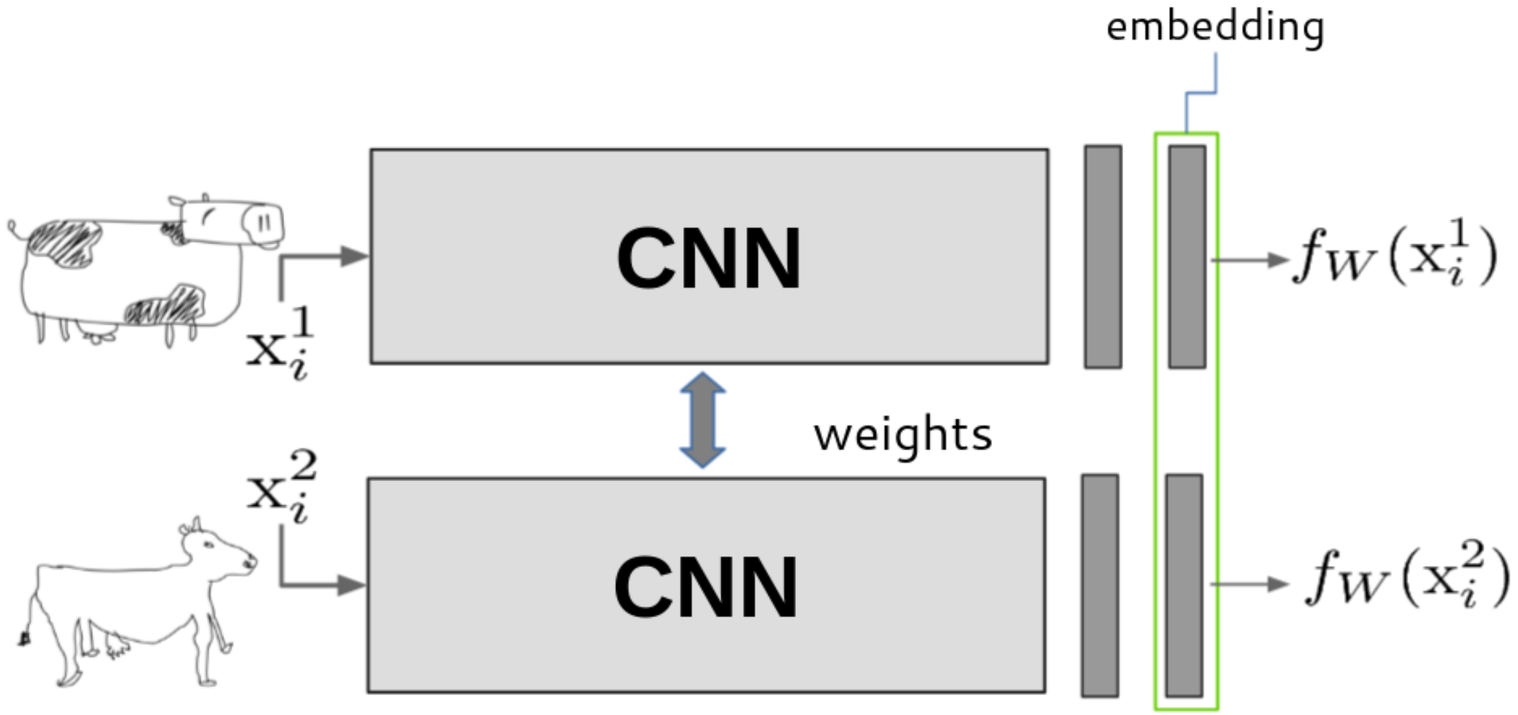


A convolutional network can learn general features that can be used in other contexts

Representation Learning

Siamese Networks

Learning a feature space (representations)

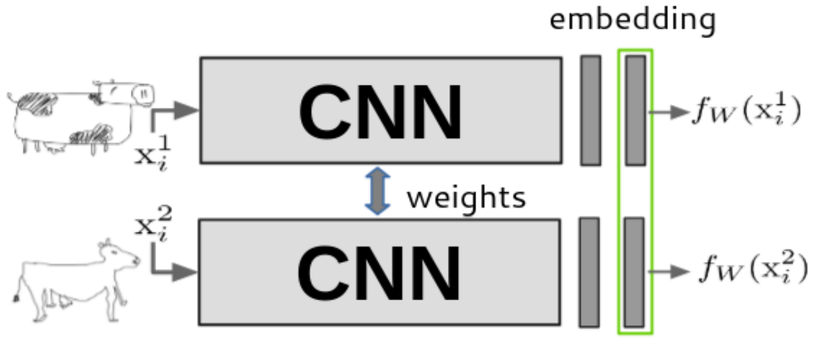


https://github.com/jmsaavedrar/siamese_networks

Representation Learning

Siamese Networks

Learning a feature space (representations)



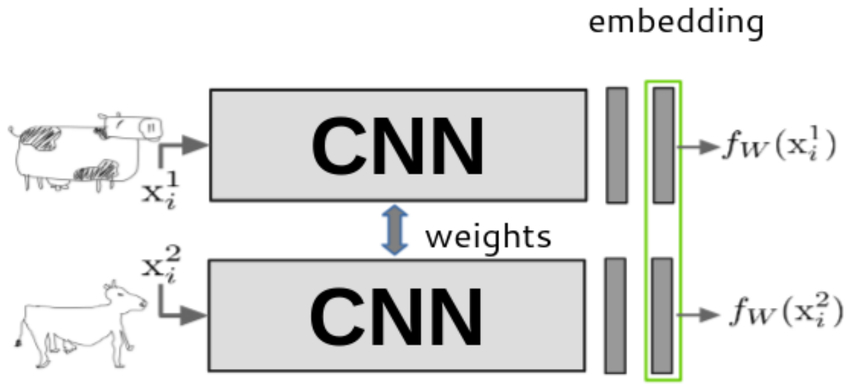
- La entrada se realiza de a pares (x_i^1, x_i^2)
- Cada para tiene asociado una etiqueta [1: similares (positivo), 0 : diferentes (negativo)]
  $(x_i^1, x_i^2) \rightarrow y_i$
- A diferencia de la clasificación, el modelo basado en *siameses* se enfoca en buscar *embeddings* que maximicen la semejanza entre objetos similares y minimice la semejanza entre objetos diferentes.

https://github.com/jmsaavedrar/siamese_networks

Representation Learning

Siamese Networks

Learning a feature space (representations)



Contrastive Loss

$$D_w^i = L_2(f_w(x_i^1), f_w(x_i^2))$$

$$L_i = y_i {D_w^i}^2 + (1 - y_i)\{max(0, \lambda - D_w^i)\}^2$$

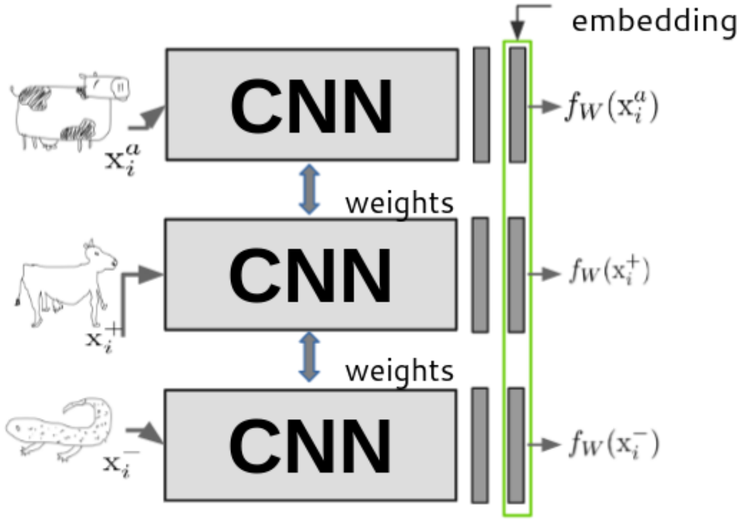
λ : margen

https://github.com/jmsaavedrar/siamese_networks

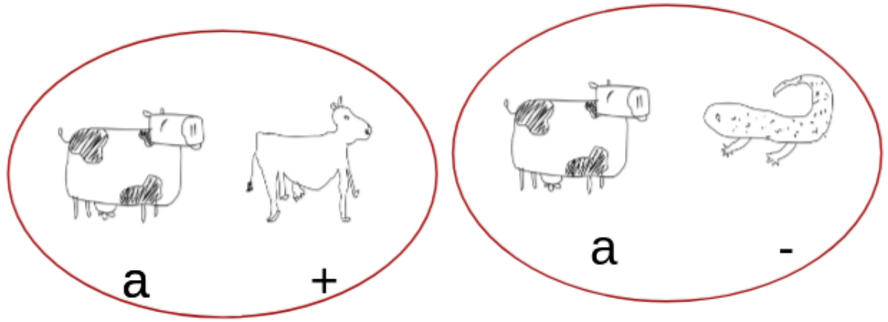
Representation Learning

Siamese Networks

Triplet Loss



[1] O. M. Parkhi, A. Vedaldi, A. ZissermanDeep Face Recognition British Machine Vision Conference, 2015



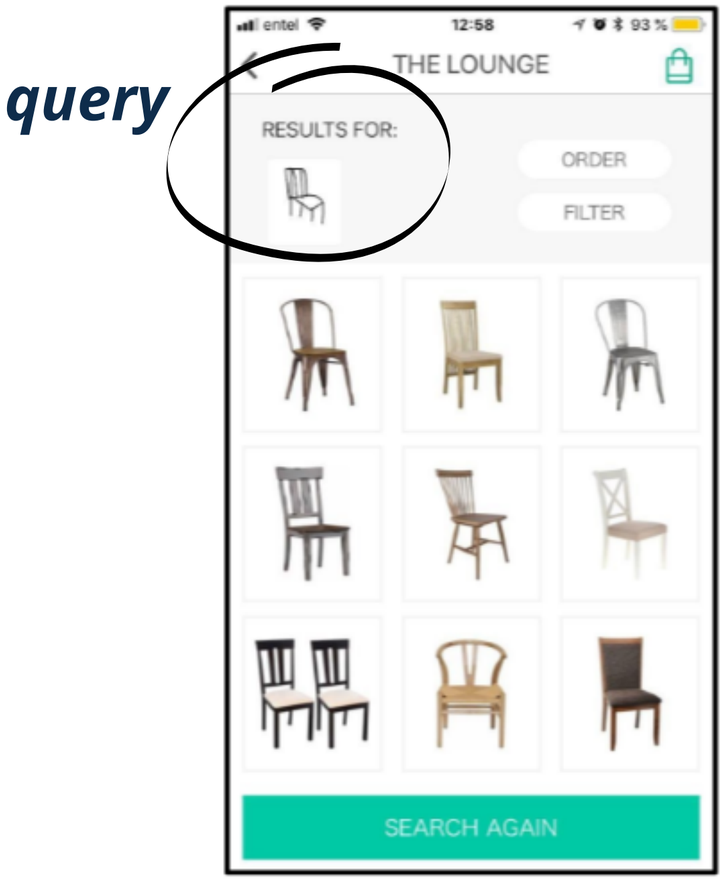
$$D_W^i(a,+) \qquad D_W^i(a,-)$$

$$D_W^i(a,+) + \lambda < D_W^i(a,-)$$

$$L_i = max(D_W^i(a,+) - D_W^i(a,-) + \lambda, 0)$$

https://github.com/jmsaavedrar/siamese_networks

Similarity Search

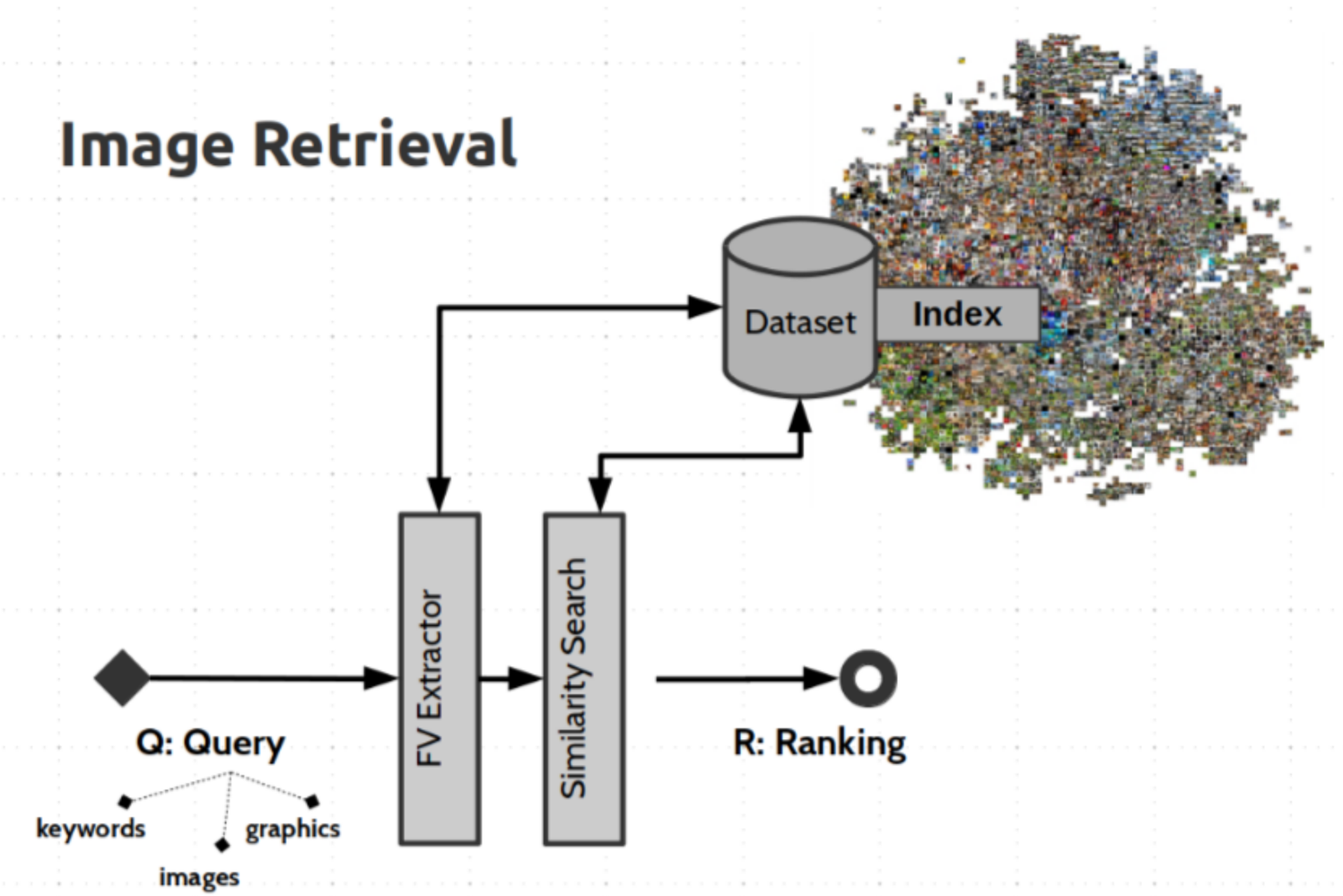


[content-based image retrieval]

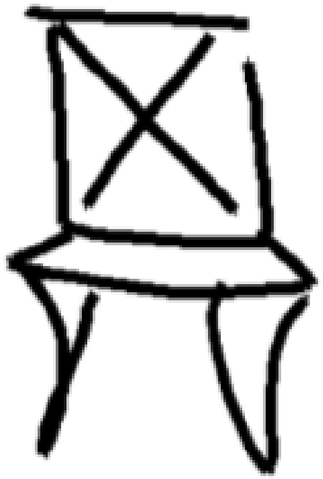
Similarity Search

Trademark Retrieval

Similarity Search



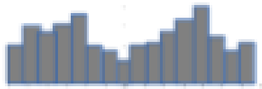
Similarity Search



$$\mathbf{x} \in \mathbb{R}^d$$

[feature vector]

$\mathbf{x}^S$ 

$\mathbf{x}^I$ 

We need to compare feature vectors in a feature space

Similarity Search

Sketch-based Image Retrieval

Mean Average Precision [mAP]

Jose M. Saavedra

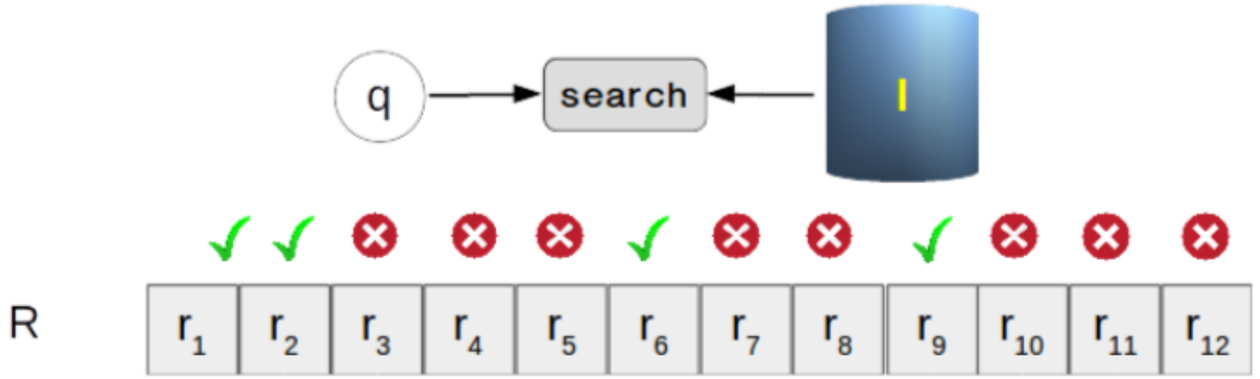
Mean Average Precision

I: Conjunto de imágenes sobre las que se busca.

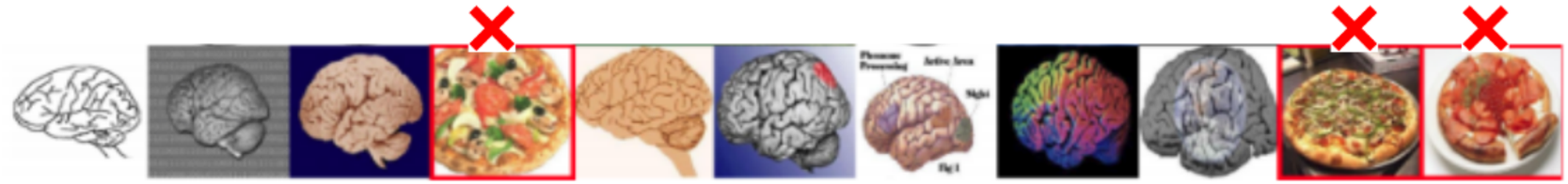
Q: Conjunto de consultas

R: Ranking, conjunto de imágenes ordenadas por similitud a una consulta

$\Gamma_q = \{x \in I, q \in Q | x \text{ es relevante para } Q\}$ conjunto de relevantes para cada q



Mean Average Precision



Mean Average Precision

Función característica, x es relevante para q

$$f_{\Gamma_q}(x) = \begin{cases} 1 & x \in \Gamma_q \\ 0 & \text{en otro caso} \end{cases}$$

Precisión (precision)

número de relevantes

$$pr_q(i; R) = \frac{\sum_{k=1}^{i} f_{\Gamma_q}(r_k)}{i} \cdot f_{\Gamma_q}(r_i)$$

Precisión indica pureza, varía entre 0 y 1 indicando mínima y máxima pureza, respectivamente.

precisión se mide solamente en relevantes

número de recuperados

Mean Average Precision

Precisión promedio para q (average precision)

$$AP_q(R) = \frac{\sum_{i=1}^{|R|} pr_q(i, R)}{\sum_{i=1}^{|R|} f_{\Gamma_q}(r_i)} \qquad AP_q(R) = \frac{\sum_{i=1}^{|R|} pr_q(i, R)}{|\Gamma_q|}$$

Mean Average Precision

$$mAP = \frac{\sum_{q \in Q} AP_q(R)}{|Q|}$$

Mean Average Precision

Ejemplo

¿Cuál es el valor de AP (Average Precision)?

Mean Average Precision

Otras variantes

- Calcular el mAP mirando solamente R relevantes. Se asume que todas las consultas tienen R relevantes como mínimo. Por ejemplo, mAP para 10 relevantes. Recordemos que la cantidad de relevantes indica **recall**.
- Un caso particular es cuando nos interesa el primer relevante. Así medimos la precision at recall =1 (precision@1), esto es conocido como **reciprocal rank**.
- Calcular mAP mirando los N primeros resultados. Es decir se limita el tamaño del ranking. En este caso los relevantes que no aparecen hasta los N primeros recuperados tendrán la misma precisión, por ejemplo 1/N.
- **Una forma de mirar el desempeño del método para diferentes tasas de recuperación (recall) podemos utilizar una gráfico Recall-Precision.**

Recall-Precision

- **Precision:** Mide el porcentaje de objetos relevantes recuperados en el ranking (es una medida de pureza de resultados).
- **Recall:** Mide el porcentaje de recuperación de objetos relevantes.

$$\text{precision}_q = \frac{|R \bigcap \Gamma_q|}{|R|}$$

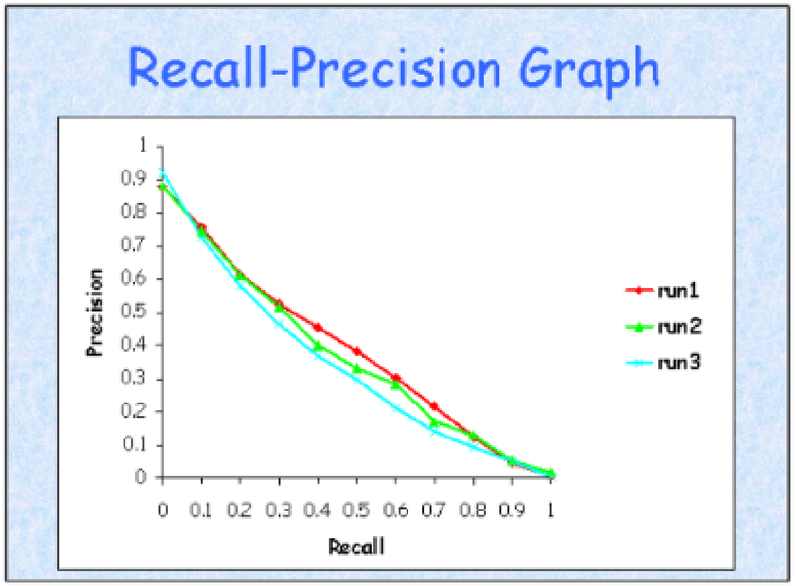
$$\text{recall}_q = \frac{|R \bigcap \Gamma_q|}{|\Gamma_q|}$$

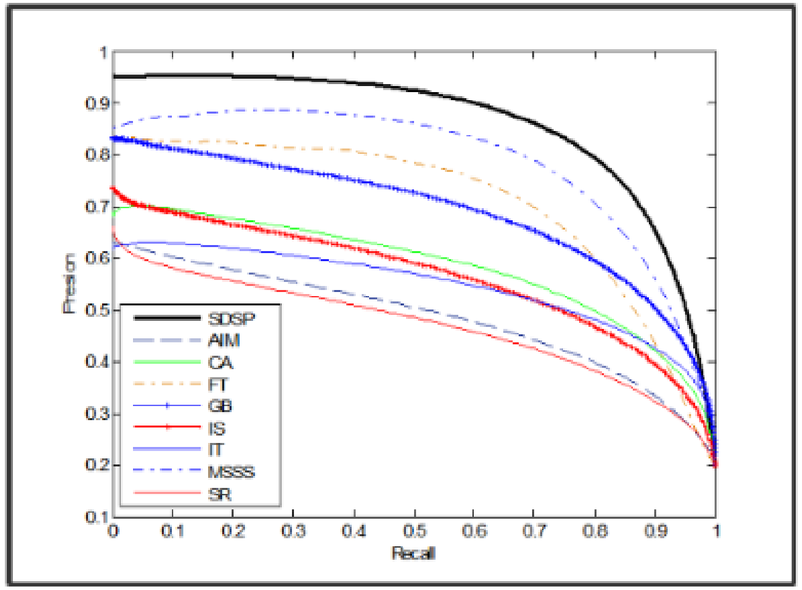
$\text{R} = \text{Ranking}$

$\Gamma_q = \text{Conjunto de relevantes para } q$

Recall-Precision

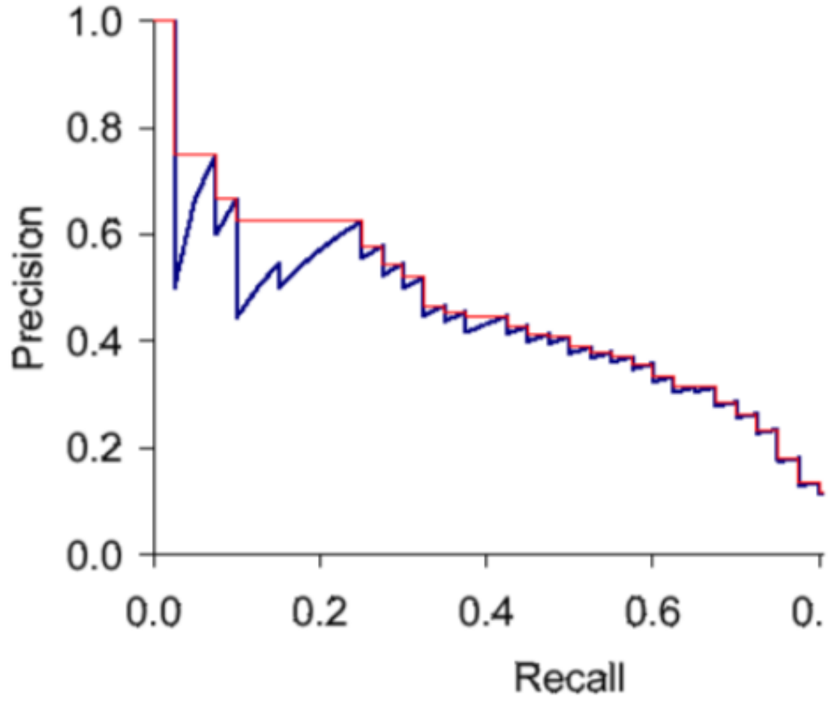
- Representa el valor de recall vs. precision
- Se consideran 11 valores de recall {0, 0.1, 0.2, 0.3, 0.4, ..., 0.9, 1}





Recall-Precision

- **Gráfico normalizado:** curva no creciente.
- **Precisión Interpolada:** La precisión para un valor de recall se calcula tomando la mayor precisión para recalls superiores.



$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

Recall-Precision

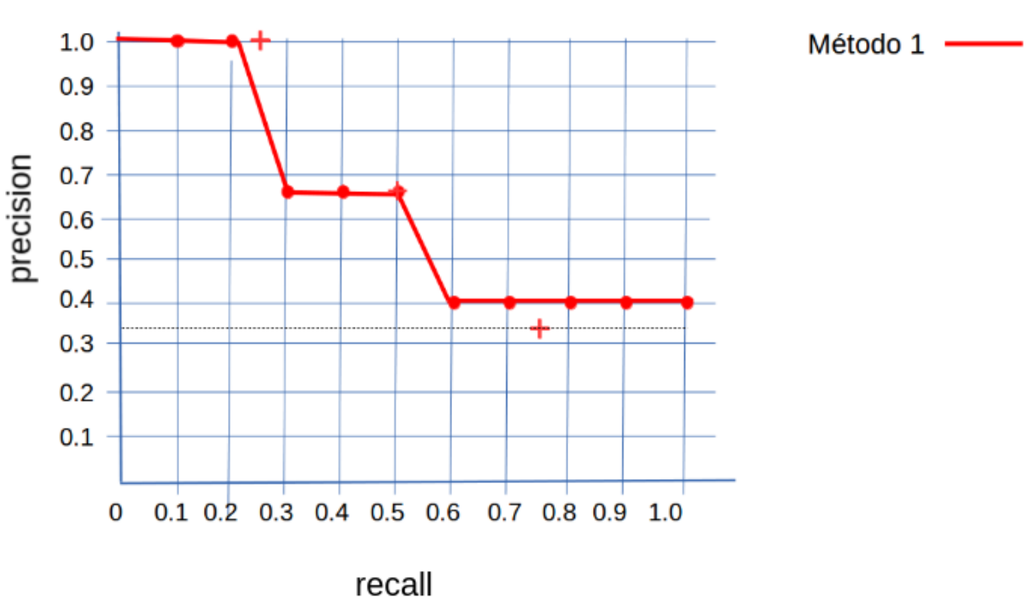
- Ejemplo (supongamos 4 relevantes y un total de 10 imágenes)
  - Método 1: R N R N N N N N R R
  - Método 2: N R N N R R R N N N

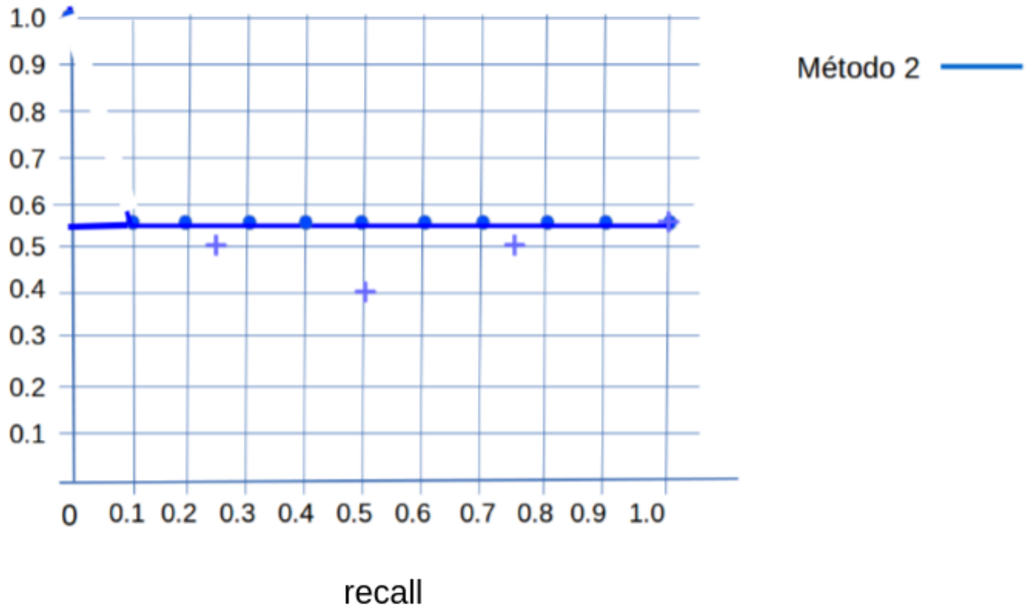
  R: Relevante N: No Relevante

Recall	Pr (Método 1)	Pr (Método 2)
1/4 = 0.25	1	1/2 = 0.5
2/4 = 0.50	2/3 = 0.67	2/5 = 0.4
3/4 = 0.75	3/9 = 0.33	3/6 = 0.5
4/4 = 1.00	4/10 = 0.4	4/7 = 0.57

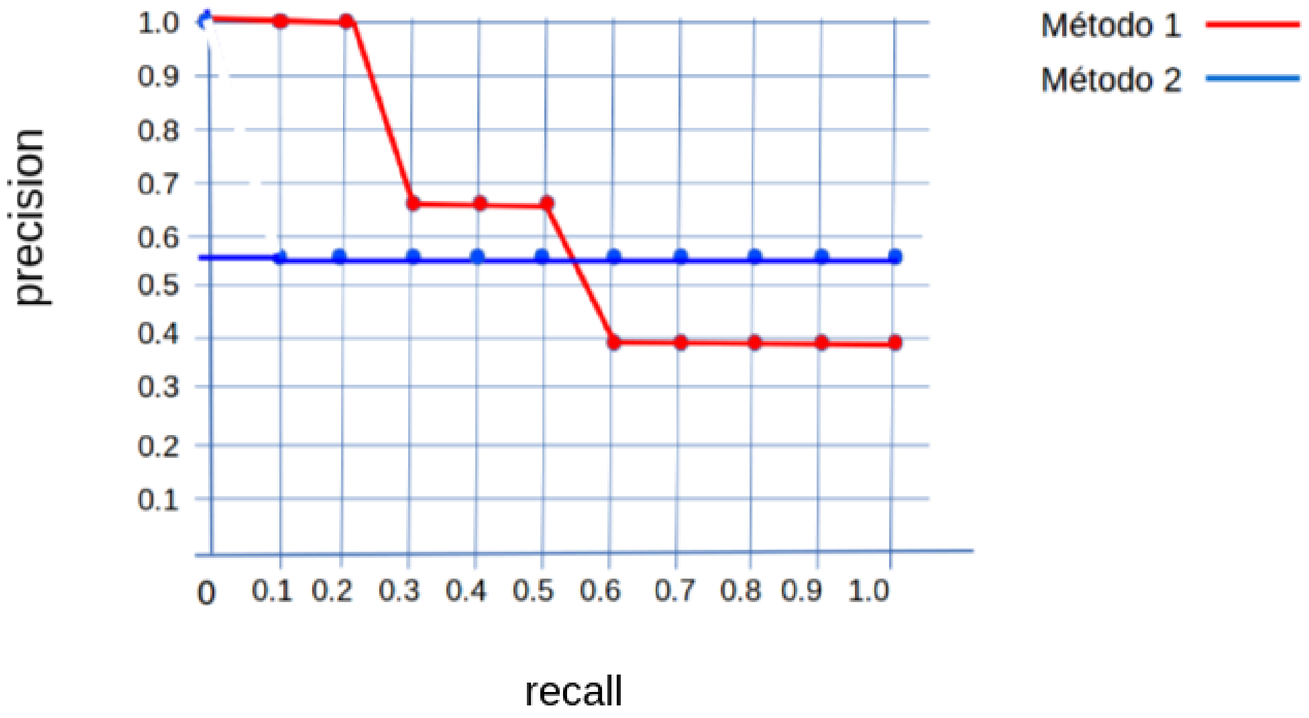
AP(método 1) = 0.6
AP(método 2) = 0.49

Recall-Precision





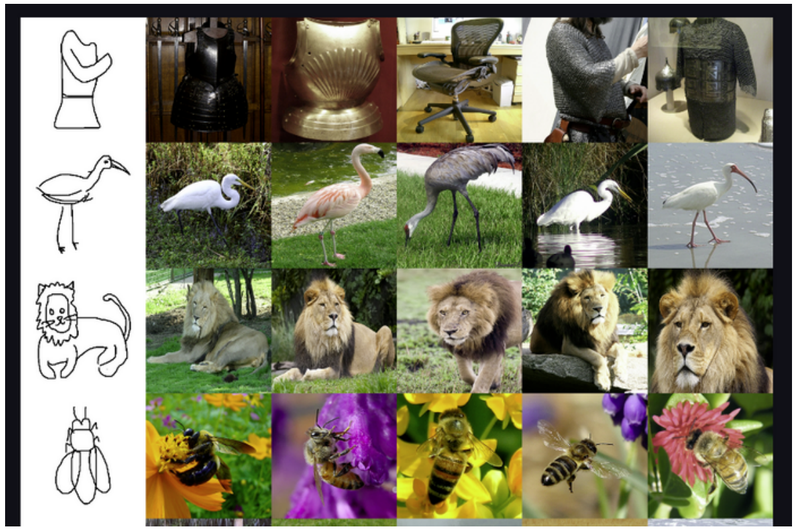
Recall-Precision



Tarea 2

Similarity Search (Siamese Networks)

- Extender la implementación de https://github.com/jmsaavedrar/siamese_networks para resolver el problema de Sketch-based Image Retrieval.
- Para lo anterior, deberán crear dos encoders, uno para procesar sketches y el otro para imágenes regulares.
- Entrenar y evaluar usando el dataset Sketchy https://github.com/CDOTAD/SketchyDatabase



Clase 5

- Self-Supervised Representation Learning