

Transferencia de Conocimiento & Vision Image Transformers

Juan Carlos Arbeláez

Agenda

1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller



Agenda

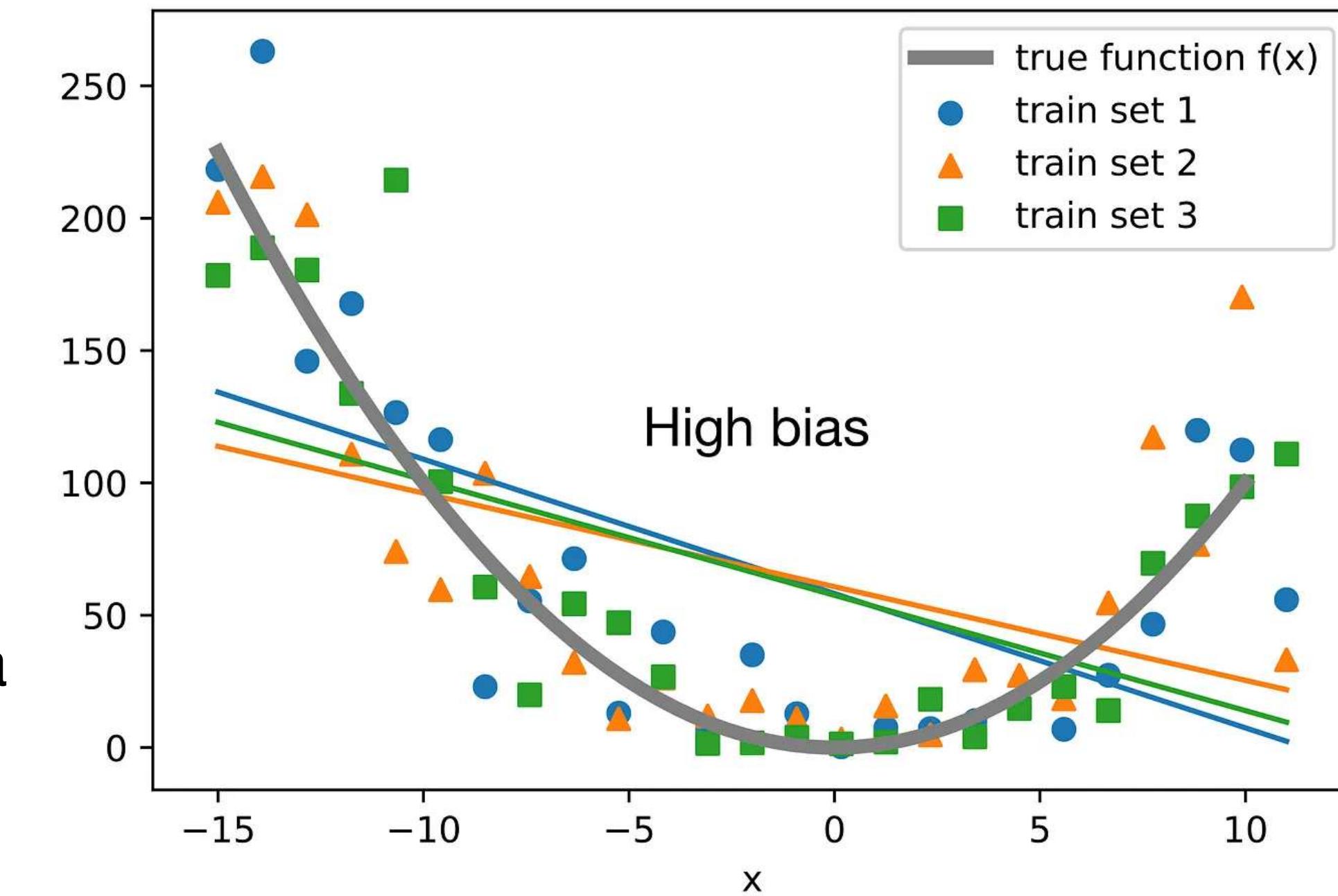
- 1. CNN sesgo inductivo**
2. Vision Image Transformers (ViT)
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller



Inductive bias

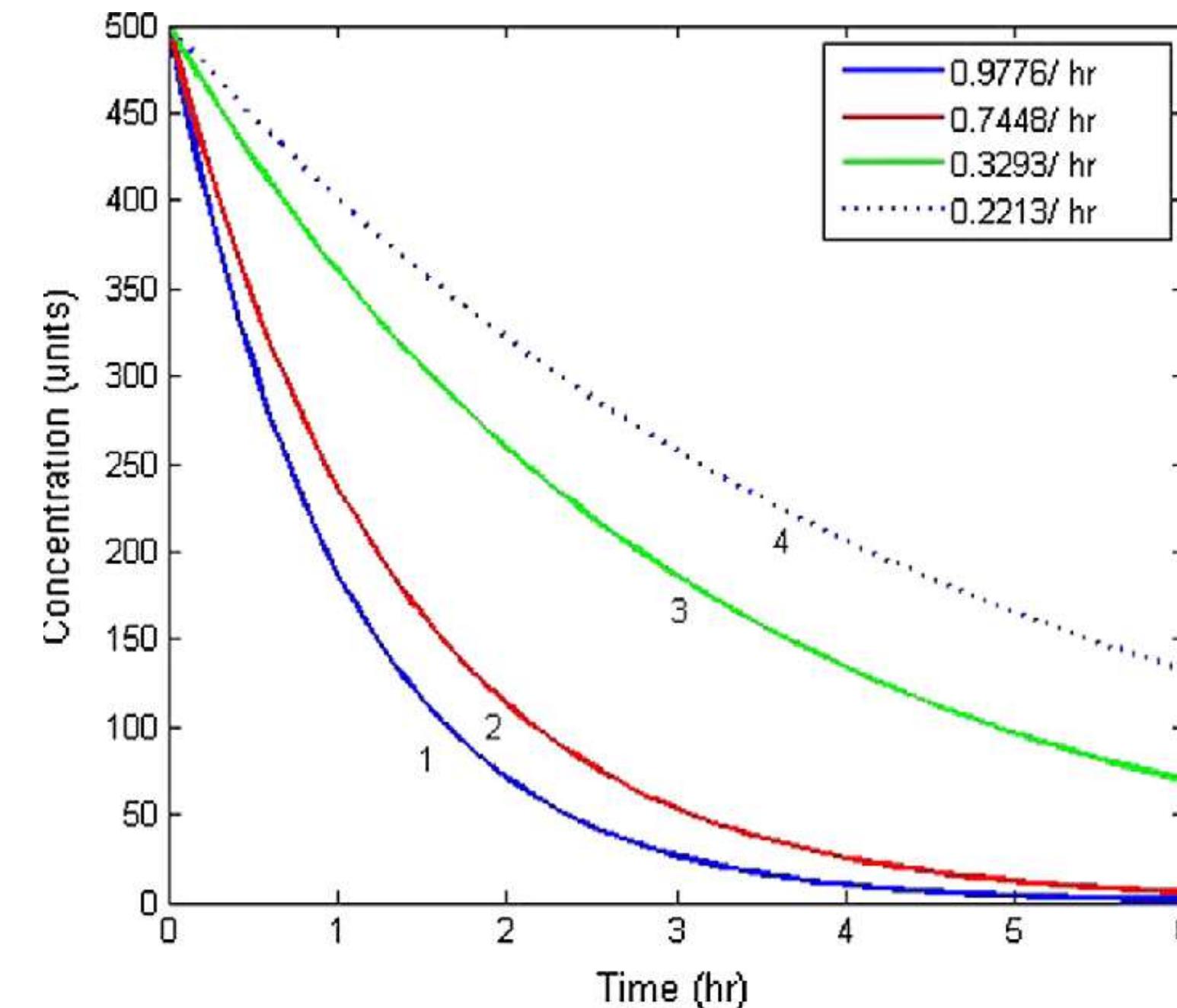
Bias

- El sesgo estadístico es un término que asociamos con el “under-fitting”
- Significa desviarse del valor real por inclinarse hacia algo. Es un error sistemático (desplazamiento de la función subyacente)
- Puede deberse a que el modelo no tiene la complejidad suficiente para ajustarse con precisión a los datos



Inductive Bias

- Es una preferencia previa debido a reglas o funciones específicas
- Por ejemplo para modelar la concentración de un medicamento se selecciona una familia de curvas exponenciales
- El sesgo inductivo no es malo si el conocimiento previo es correcto. Puede conducir a modelos más eficientes (menos número de parámetros)



Inductive Bias

Redes Neuronales Convolucionales

Hay un sesgo implícito en la arquitectura por el uso de convoluciones (filtros deslizándose por la imagen):

1. **Independencia translacional:** Una característica es igual independiente de su posición en la imagen
2. **Localidad:** Los píxeles cercanos tienen más probabilidades de estar relacionados o compartir patrones significativos que los distantes
3. **Composición:** la composición de características de paso a conceptos mas abstractos

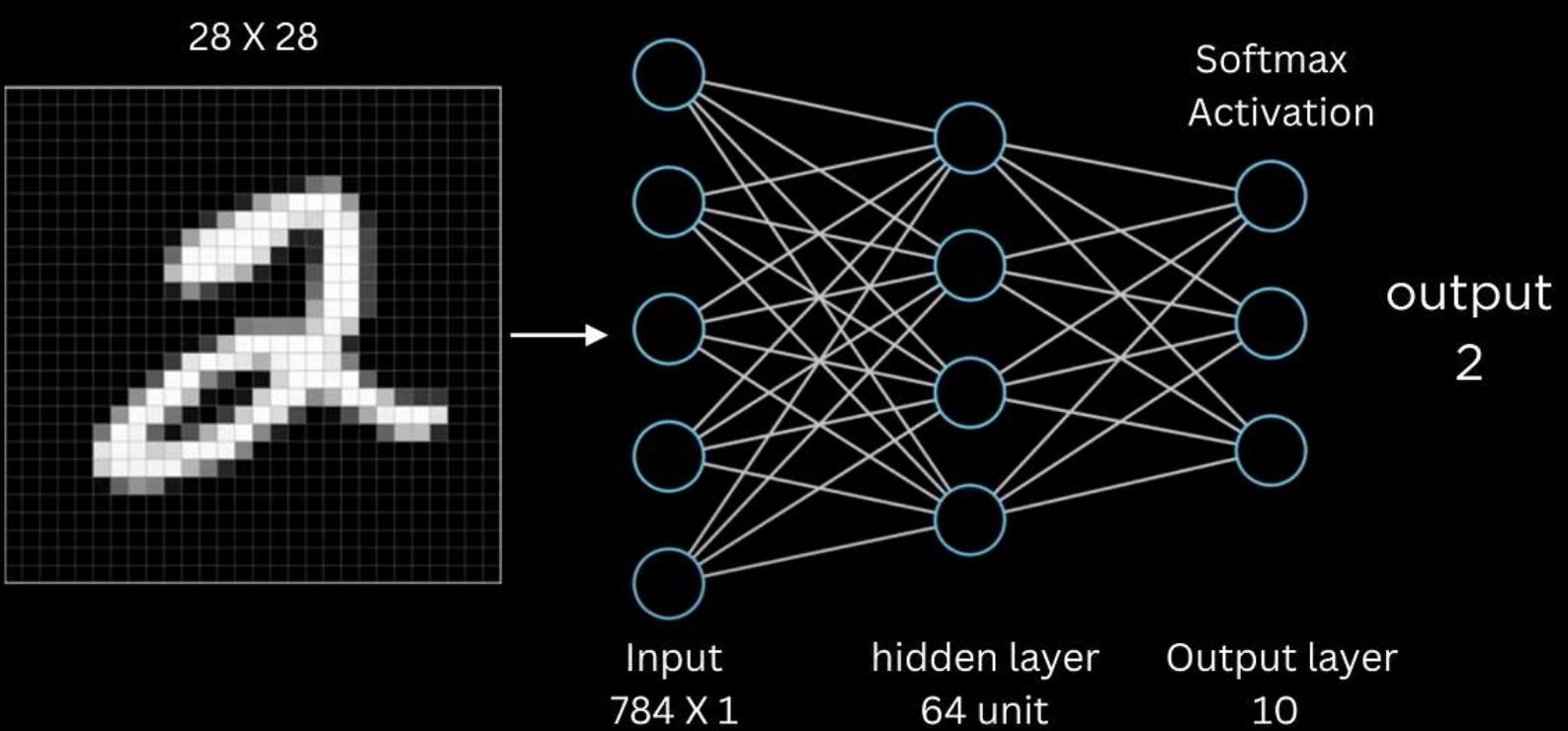


Inductive Bias

Redes Neuronales Convolucionales

Estas suposiciones vienen de nuestro conocimiento de las imágenes.

En teoría, una red neuronal podría aprender las mismas características, pero necesitaría muchos más datos, tiempo y recursos



Agenda

1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
 - A. ViT
 - B. Swin Transformer
 - C. CLIP
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller



ViT

Vision Transformer (ViT)

Una arquitectura que aplica
“Transformers” a la clasificación de
imágenes (originalmente diseñados
para procesamiento de lenguaje
natural)

Propuesto en "An Image Is Worth 16x16
Words: Transformers for Image
Recognition at Scale" (2021)

Motivación

Vision Transformer (ViT)

Las CNN tienen fuertes sesgos inductivos como:

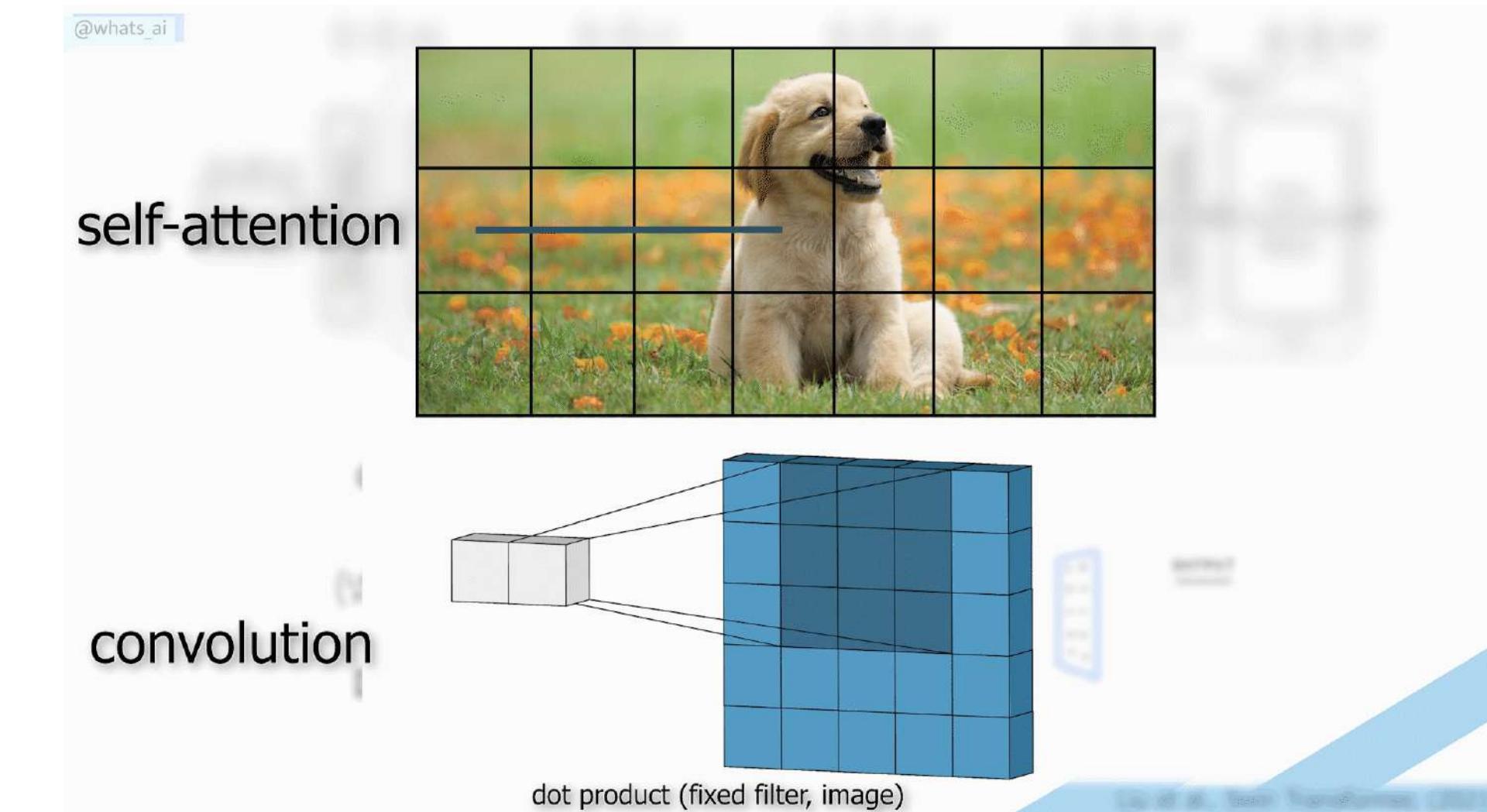
- Las convoluciones se centran en regiones pequeñas y locales
- Los pesos se comparten a través de ubicaciones espaciales

Las ViT tienen mayor capacidad de representación al no tener tanto sesgo inductivo “hardcoded” (Aprenden las relaciones espaciales)

Motivación

Vision Transformer (ViT)

- A las CNN se les dificulta capturar las relaciones entre características que están alejadas en las imágenes
- Éxito del mecanismo de “Self-Attention” para capturar relaciones globales en NLP
- Iguala o supera a las CNN en grandes conjuntos de datos
- Arquitectura unificada para problemas de visión y lenguaje (Modelos multimodales)

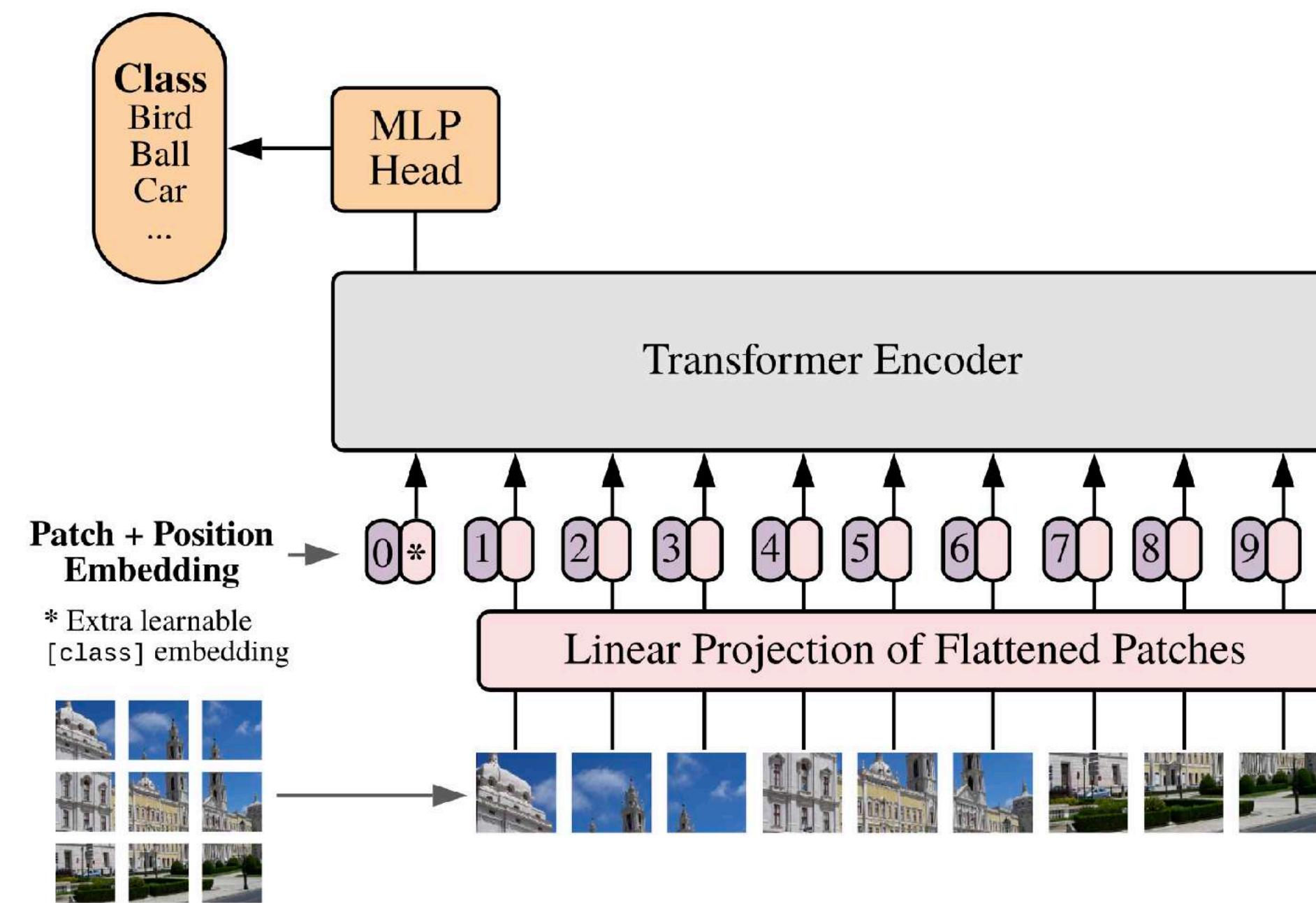


Arquitectura

Vision Transformer (ViT)

Elementos clave:

- A. División en “patches”
- B. “Patch Embedding”
- C. “Positional Embedding”
- D. “Transformer Encoder”
- E. [CLS] Token



Patch Embedding

Vision Transformer (ViT)

- La imagen de entrada se divide en “patches” cuadrados de tamaño fijo
- Cada parte se aplana y se pasa por una capa densa compartida (“learnable linear projection”)
- El resultado es una secuencia de parches embebidos

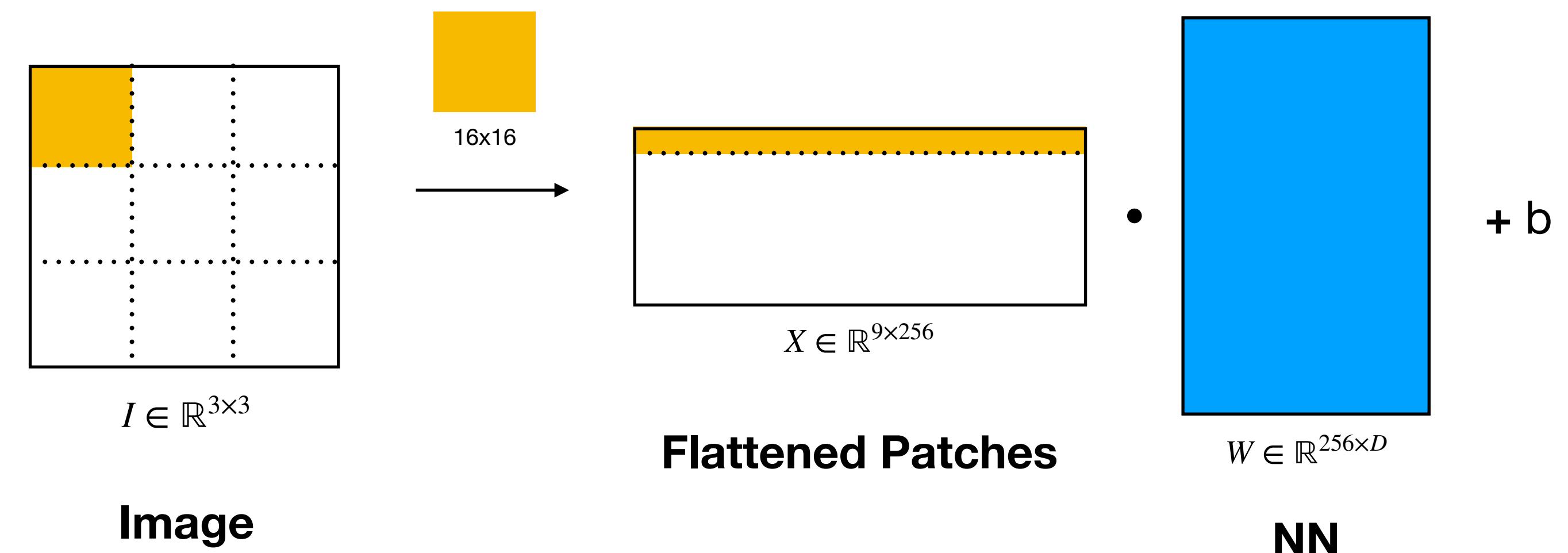
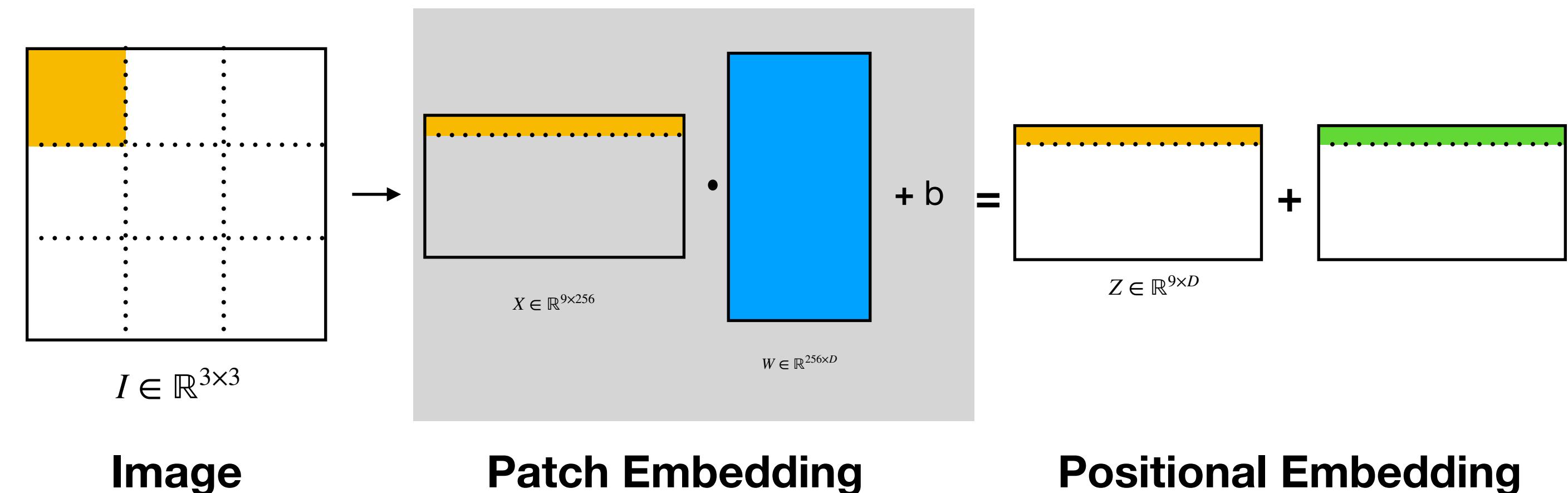


Fig: Ejemplo en imagen 2d

Positional Embedding

Vision Transformer (ViT)

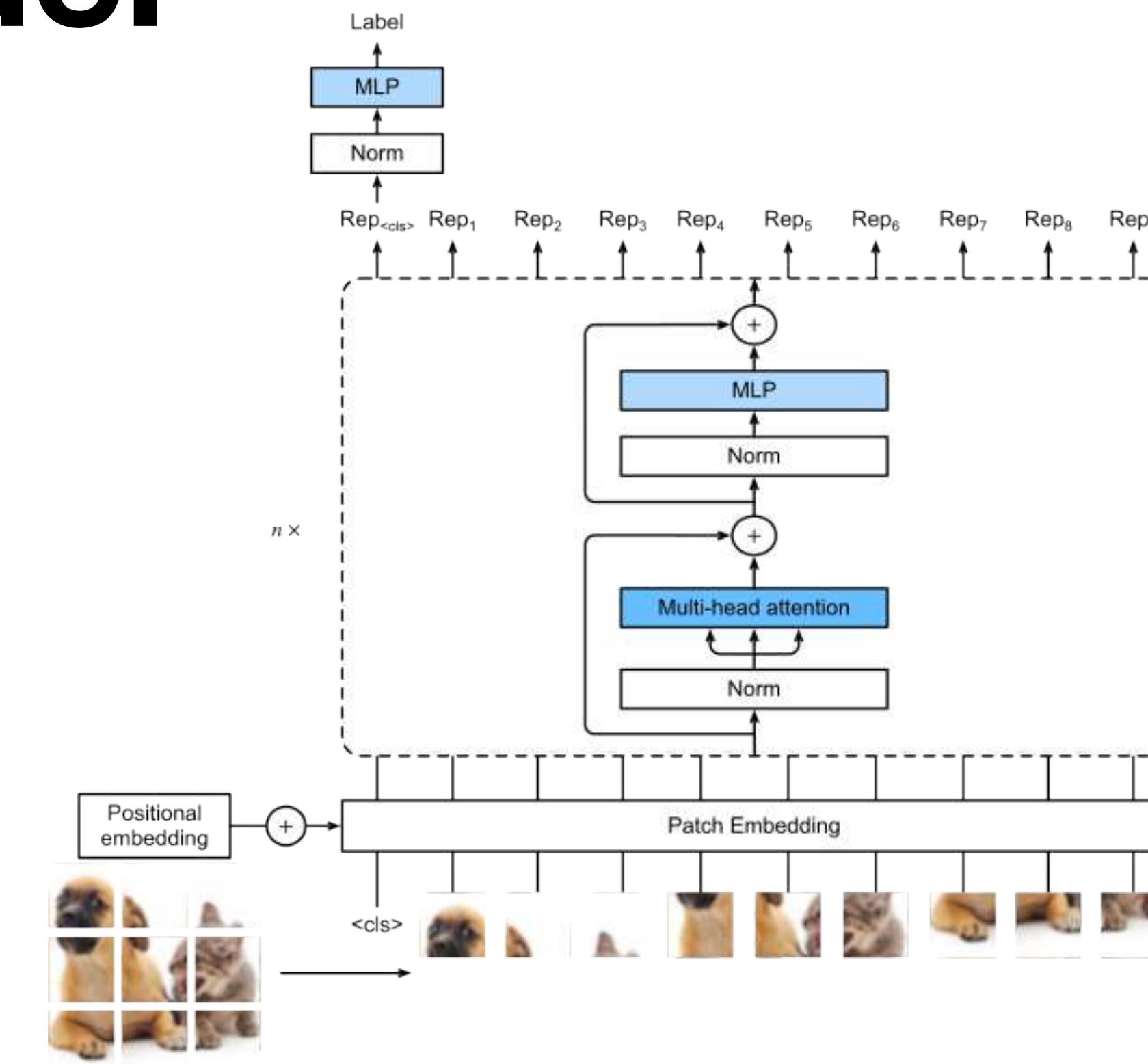
- Los “Transformers” carecen de entendimiento espacial inherente
- La información de posición de las características debe proveerse de forma explícita
- Consiste en una matriz global (que se entrena) que se suma a los “patches”



Transformer Encoder

Vision Transformer (ViT)

- Procesa una secuencia “patches” para aprender una representación contextual
- Captura globalmente las relaciones entre los “patches”
- Compuesto de multiples capas idénticas compuestas de:
 - Multi-head self attention (MSA)
 - Capa de normalización
 - Capa densa
 - Conexiones residuales

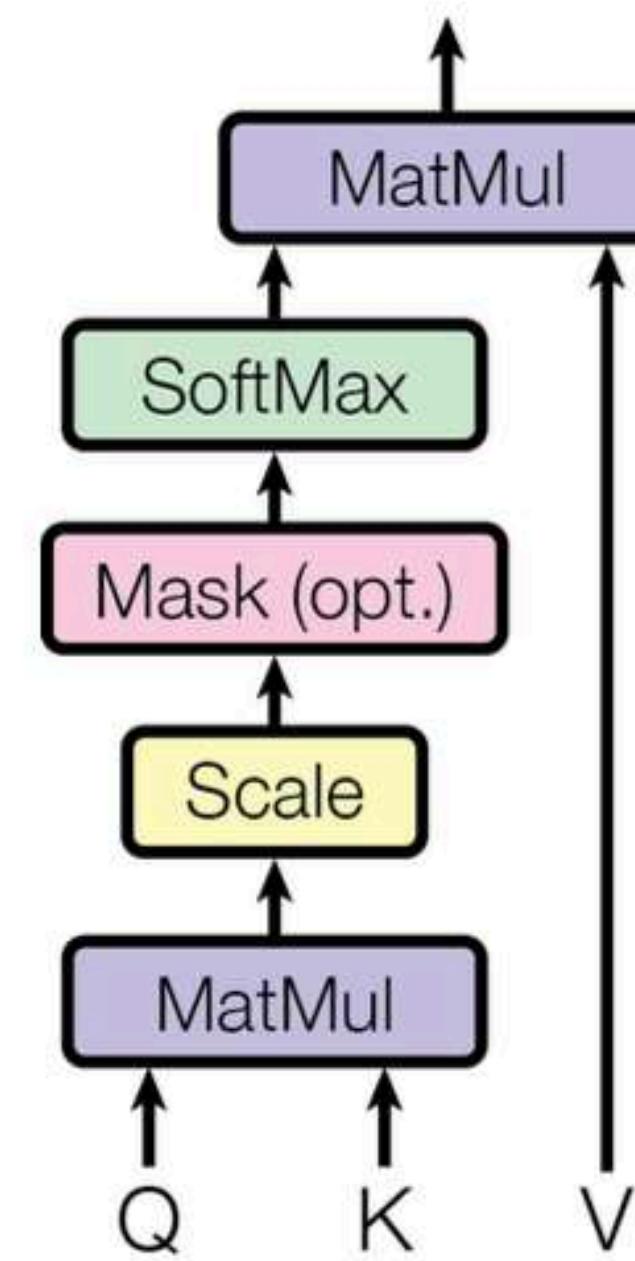


MSA

Multi-head self attention

- Calcula un mapa de atención (A): cuánto debe prestar atención cada embebido a todos los demás
- En O se mezcla la información de los tokens según la atención aprendida
- Se calcula una nueva representación ponderando cada embebido con el mapa de atención

Scaled Dot-Product Attention

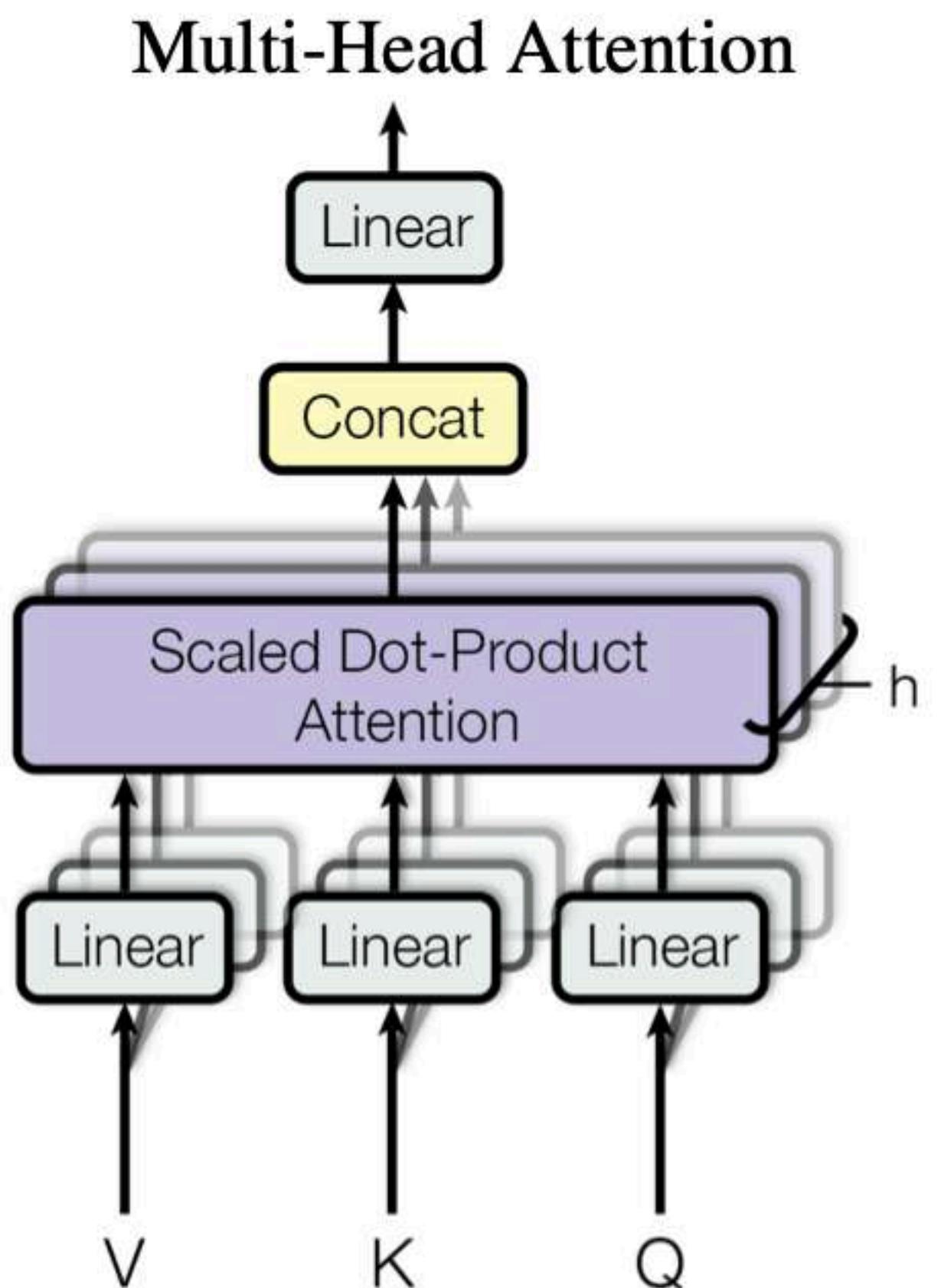


$$A = \text{Softmax} \left(\frac{QK^T}{\sqrt{d_K}} \right)$$
$$O = A \cdot V$$

MSA

Multi-head self attention

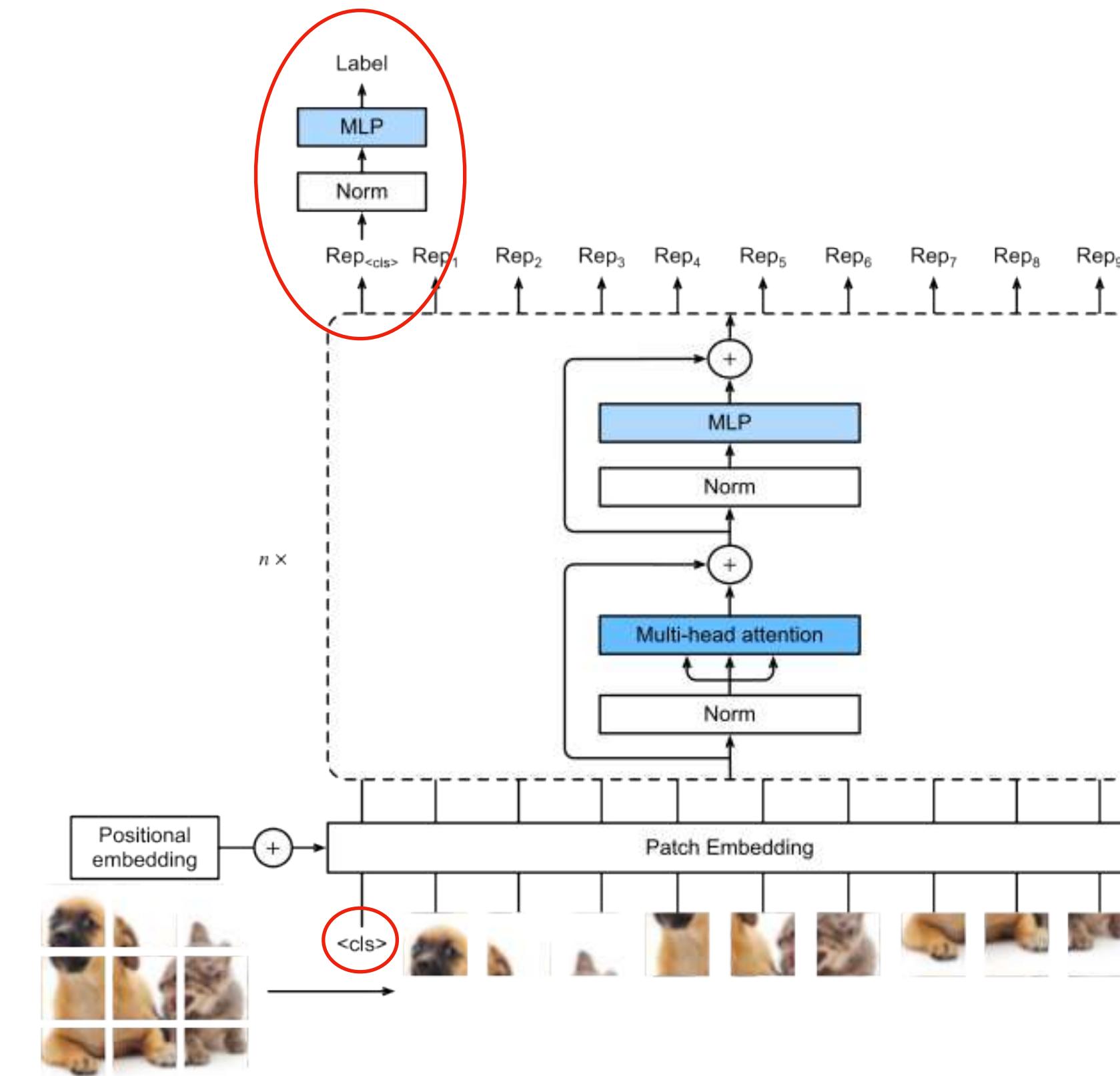
- El proceso anterior se repite en paralelo para varias cabezas, cada uno con sus propias matrices de proyección
- Cada cabezal puede capturar diferentes tipos de relaciones (local frente a global, forma frente a textura)
- Los resultados se concatenan y se proyectan de nuevo a la dimensión original



[CLS] Token

Vision Transformer (ViT)

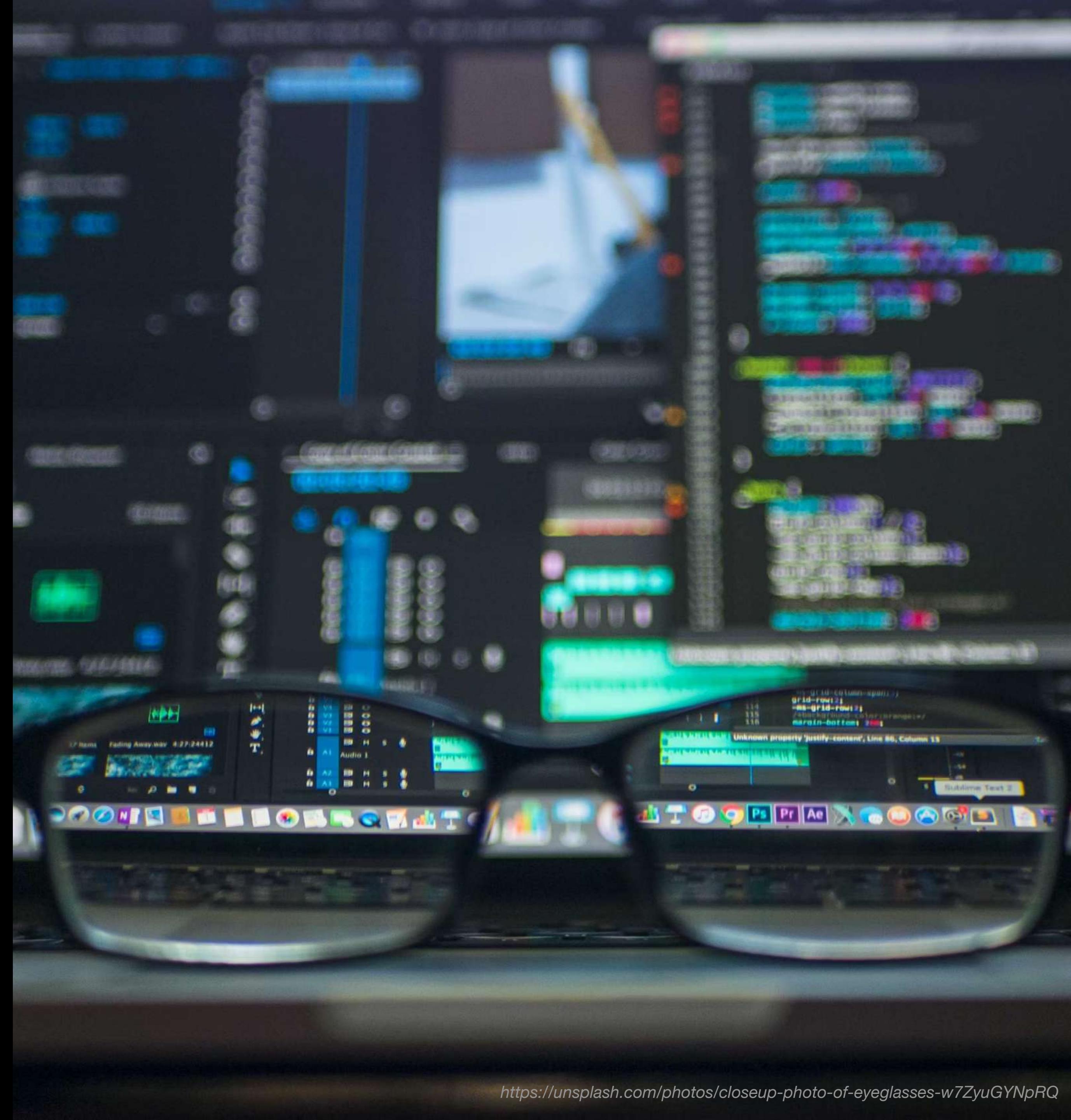
- A la secuencia de parches se le añade un “token” adicional [CLS]
- En este “token” se agrega información de todos los parches durante el proceso de transformación y se utiliza para la clasificación
- Actúa como un representación resumida de toda la imagen y es el la entrada a la cabeza de clasificación



Aplicaciones

Vision Transformer (ViT)

- Clasificación de imágenes (ViT)
- Detección de objetos (DETR)
- Segmentación de imágenes
- Modelos generativos
- Modelos Multi-modales (CLIP)



Limitaciones

Vision Transformer (ViT)

Bajo rendimiento en conjuntos de datos pequeños: requiere grandes conjuntos de datos para superar a las CNN

Alto coste computacional: aumenta cuadráticamente con el tamaño de la entrada



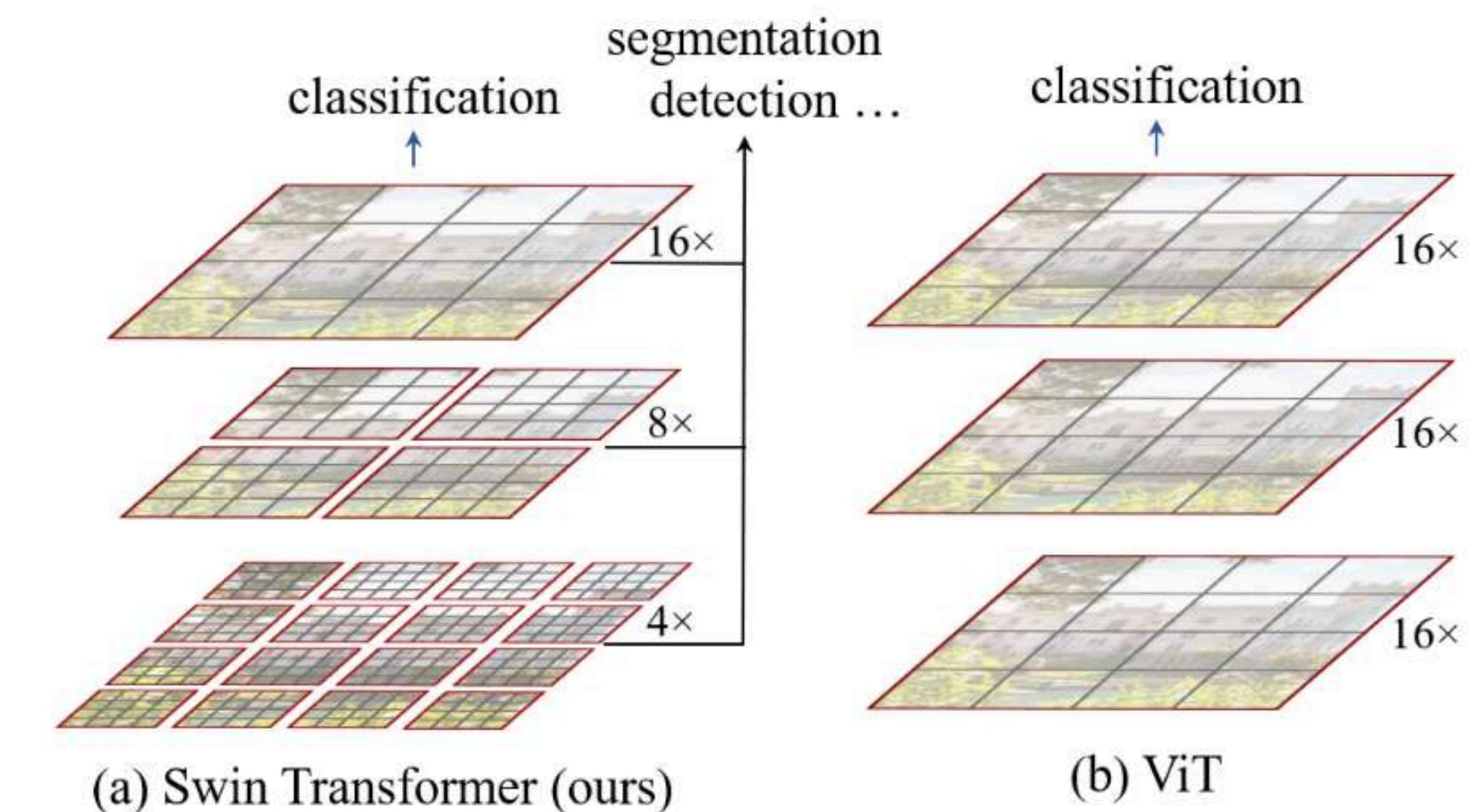
Agenda

1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
 - A. ViT
 - B. Swin Transformer**
 - C. CLIP
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller



Swin Transformers

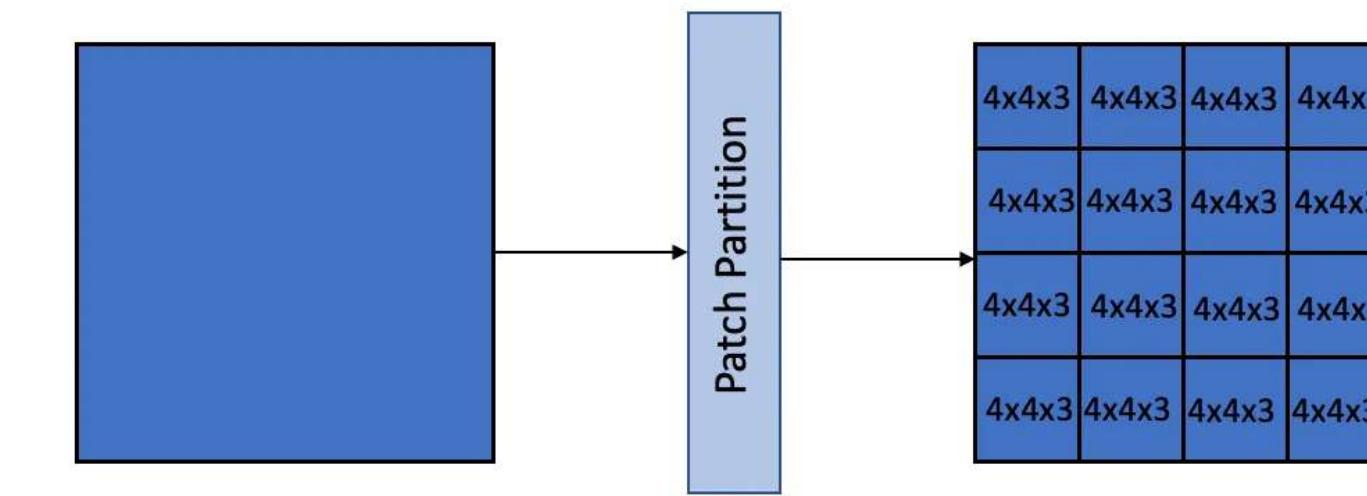
- Son un tipo de ViT enfocadas en mejorar la eficiencia y la escalabilidad
- Los ViT tienen dificultades con las imágenes de alta resolución (alto coste computacional)
- Swin Transformers introduce 2 conceptos claves: “***hierarchical feature maps***” y “***shifted window attention***”



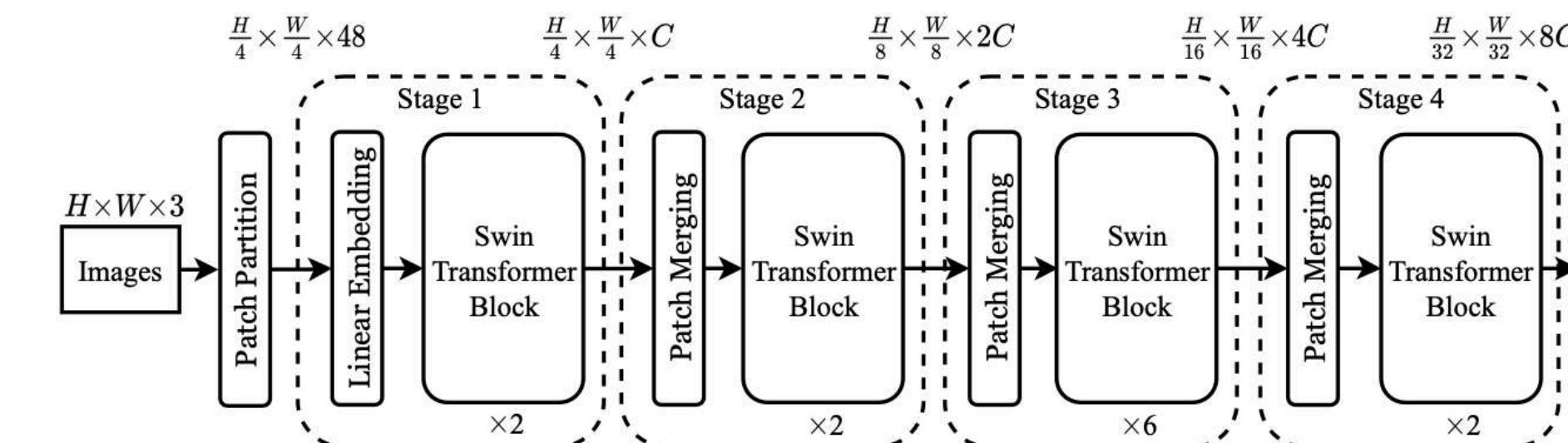
Swin Transformers

Mapas de características jerárquicos

- La imagen es dividida y los patches son embebidos como en ViT
- En cada etapa del modelo se producen “feature maps” que son representaciones intermedias de la imagen
- Los “feature maps” son reducidos espacialmente a la mitad en cada etapa a través de “patch merging”



Partición de la imagen

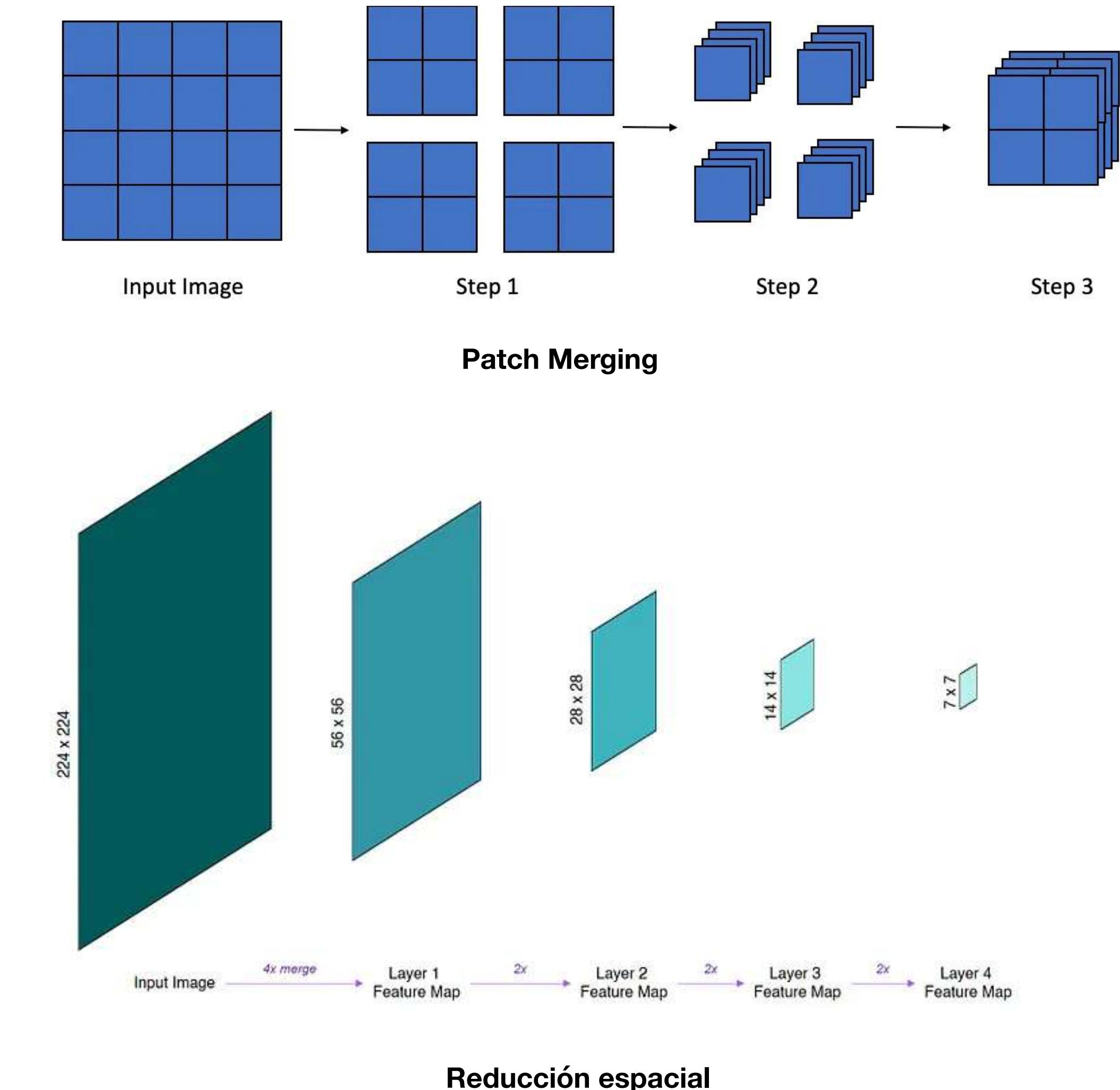


Arquitectura Swin Transformer

Swin Transformers

Mezcla de parches

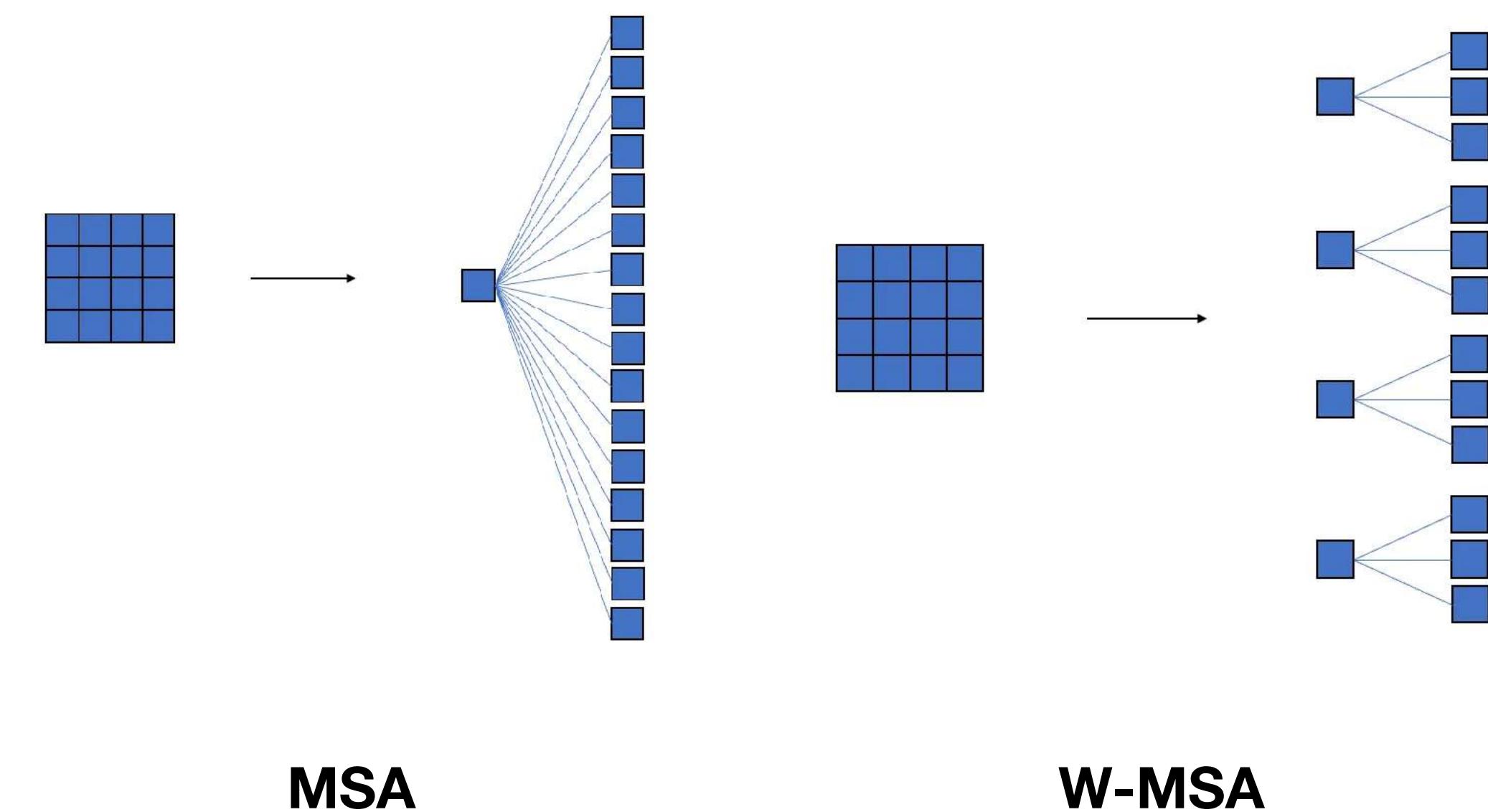
- Reducción espacial del mapa de características concatenando en profundidad
- No contribuye en mejora de la eficiencia ya que el tamaño de los mapas de características permanecen relativamente constantes
- Pero si en tener una representación jerárquica



Swin Transformers

Ventana de atención desplazada

- Atención global es ineficiente para imágenes de alta resolución
- Reemplaza con ***Window-based*** y ***Shifted-Window MSA***
- *Self-attention* es calculado en ventanas locales que no se sobreponen



MSA

W-MSA

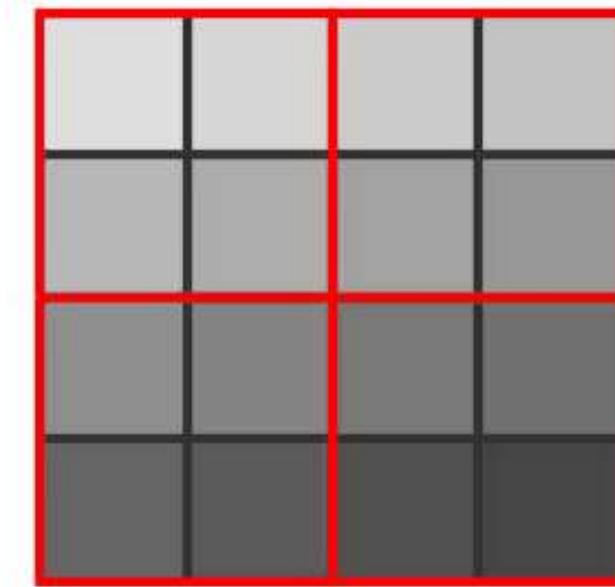
Swin Transformers

Ventana de atención desplazada

- Atención en ventanas carece de conexiones entre ventanas
- Shifted-Window-MSA desplaza las ventanas hacia la esquina inferior derecha
- Aplica una técnica de desplazamiento cíclico que desplaza los parches huérfanos a ventanas con parches incompletos
- Este enfoque permite conexiones cruzadas entre ventanas

Shifted Window MSA

Step 1: Shift window by a factor of $M/2$, where M = window size
Step 2: For efficient batch computation, move patches into empty slots to create a complete window.
This is known as 'cyclic shift' in the paper.



Agenda

1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
 - A. ViT
 - B. Swin Transformer
 - C. CLIP**
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller



Contrastive Language-Image Pre-training

- Modelo multimodal que combina visión y lenguaje
- Publicado por OpenAI en *Learning Transferable Visual Models From Natural Language Supervision* (2021)
- Capaz de realizar transferencia zero-shot en aplicaciones downstream



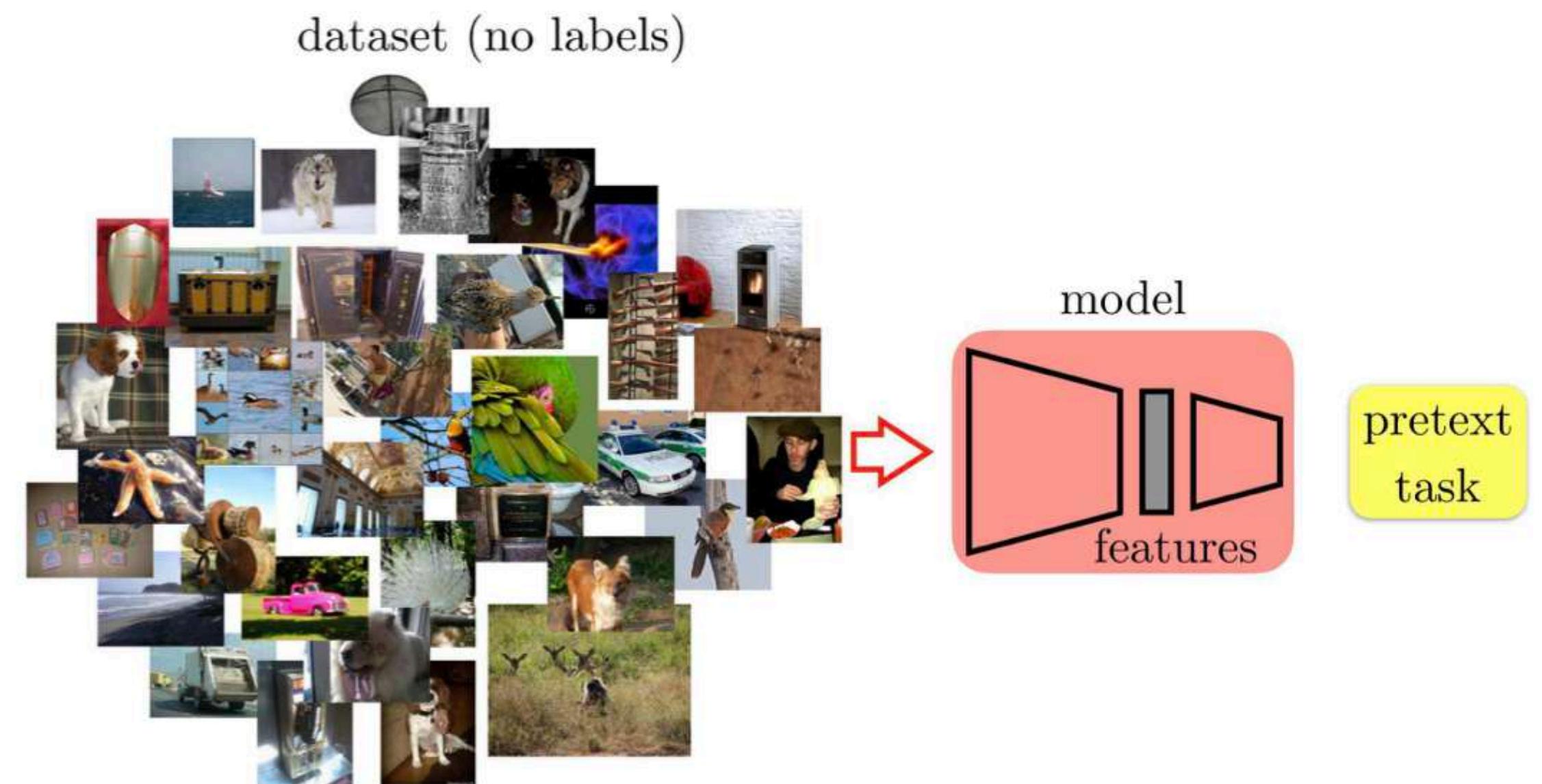
Motivación

Entrenamiento autosupervisado en LLMs:

- Pre-entrenamiento a gran escala gracias arquitecturas y objetivos agnósticos
- Transferencia de aprendizaje *zero-shot*

Pero en CV:

- Pre-entrenado en grandes conjuntos etiquetados (ImageNet)
- Barrera en la generalización en tareas no pre-entrenadas





CLIP

Aprender la percepción desde la supervisión del lenguaje natural

1. Conjunto de datos
2. Aprendizaje contrastivo
3. Arquitectura

CLIP

Conjunto de datos

- Supervisión por lenguaje natural se beneficia de grandes disponibilidades de datos en internet
- En visión no hay datasets de imágenes con descripciones de calidad suficiente
- Dataset de 400 million pares de (imagen, texto) de fuentes públicas
- Se balanceo el conjunto con búsqueda de imágenes en internet de 500K queries



The man at bat readies to swing at the pitch while the umpire looks on.



A large bus sitting next to a very tall building.



A horse carrying a large load of hay and two people sitting on it.



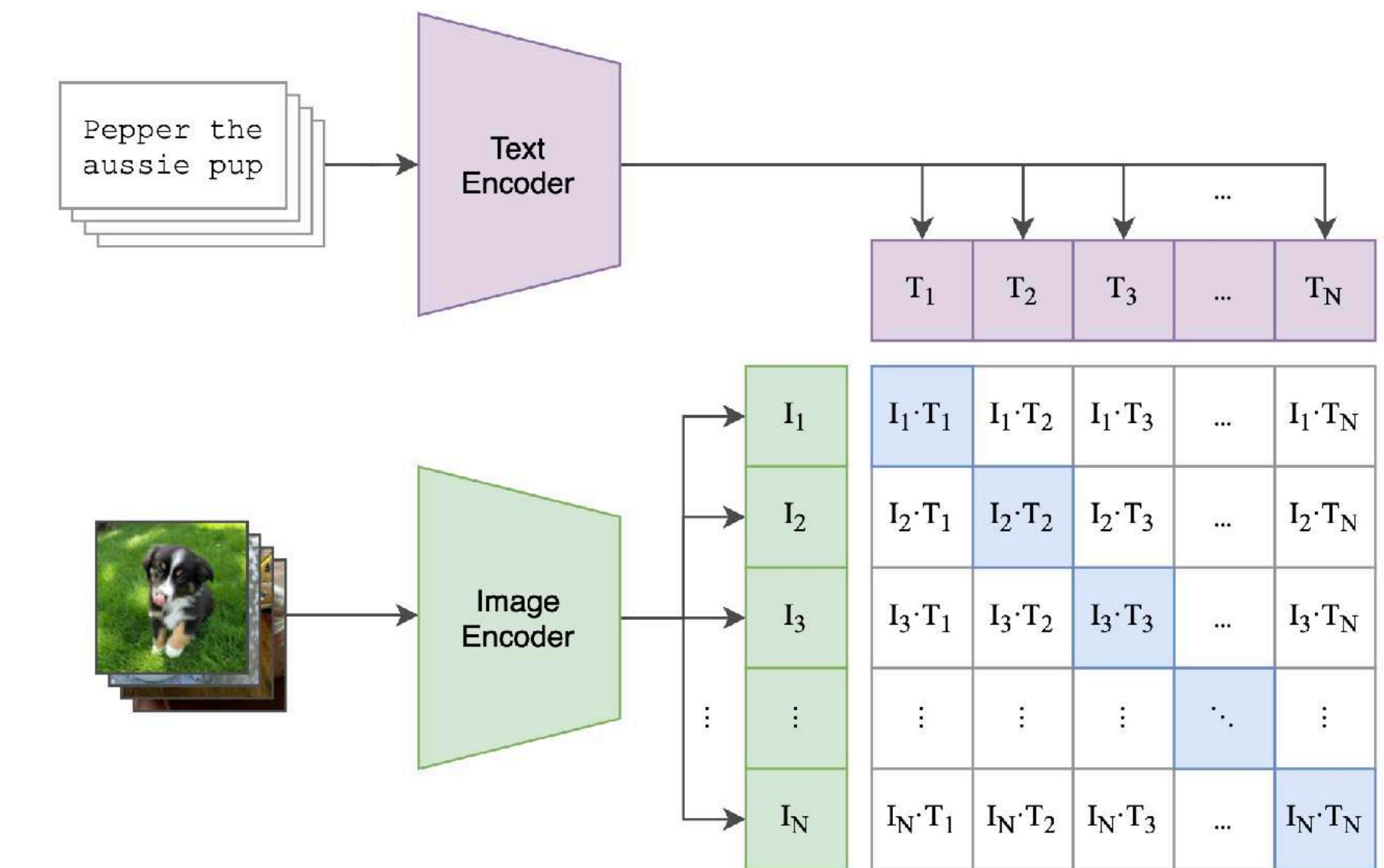
Bunk bed with a narrow shelf sitting underneath it.

Fig. 1: Example images and captions from the Microsoft COCO Caption dataset.

CLIP

Aprendizaje contrastivo

- Predecir qué texto está emparejado con cuál imagen
- Se entrena conjuntamente los encoders (texto e imagen)
- Maximizar la similitud coseno de pares de embebidos (texto-imagen) correctos y minimizar los incorrectos



CLIP

Codificación de texto

- Cada palabra es subdividida en unidades más pequeñas - *tokens* (BPE algoritmo)
- A cada *token* se le asigna un índice que apunta a un vector
- El vector es aprendido durante el entrenamiento (embebido)

Sequence of token	Index	Positional Encoding Matrix			
I	0	P_{00}	P_{01}	...	P_{0d}
am	1	P_{10}	P_{11}	...	P_{1d}
a	2	P_{20}	P_{21}	...	P_{2d}
Robot	3	P_{30}	P_{31}	...	P_{3d}

Positional Encoding Matrix for the sequence 'I am a robot'

CLIP

Aprendizaje contrastivo

Elementos de la función de costo:

1. Par correspondiente de embebidos de texto e imágenes (I_i, T_i)
2. Similitud coseno entre los embebidos
3. Optimización del cross-entropy simétrico (fila ℓ_i^{img} y columna ℓ_i^{text}) del softmax de las similitudes

$$\mathcal{L} = \frac{1}{2N} \sum_{i=1}^N [\ell_i^{\text{img}} + \ell_i^{\text{text}}]$$

Cross-entropy por fila:

$$\ell_i^{\text{img}} = -\log \frac{\exp(\text{sim}(I_i, T_i)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(I_i, T_j)/\tau)}$$

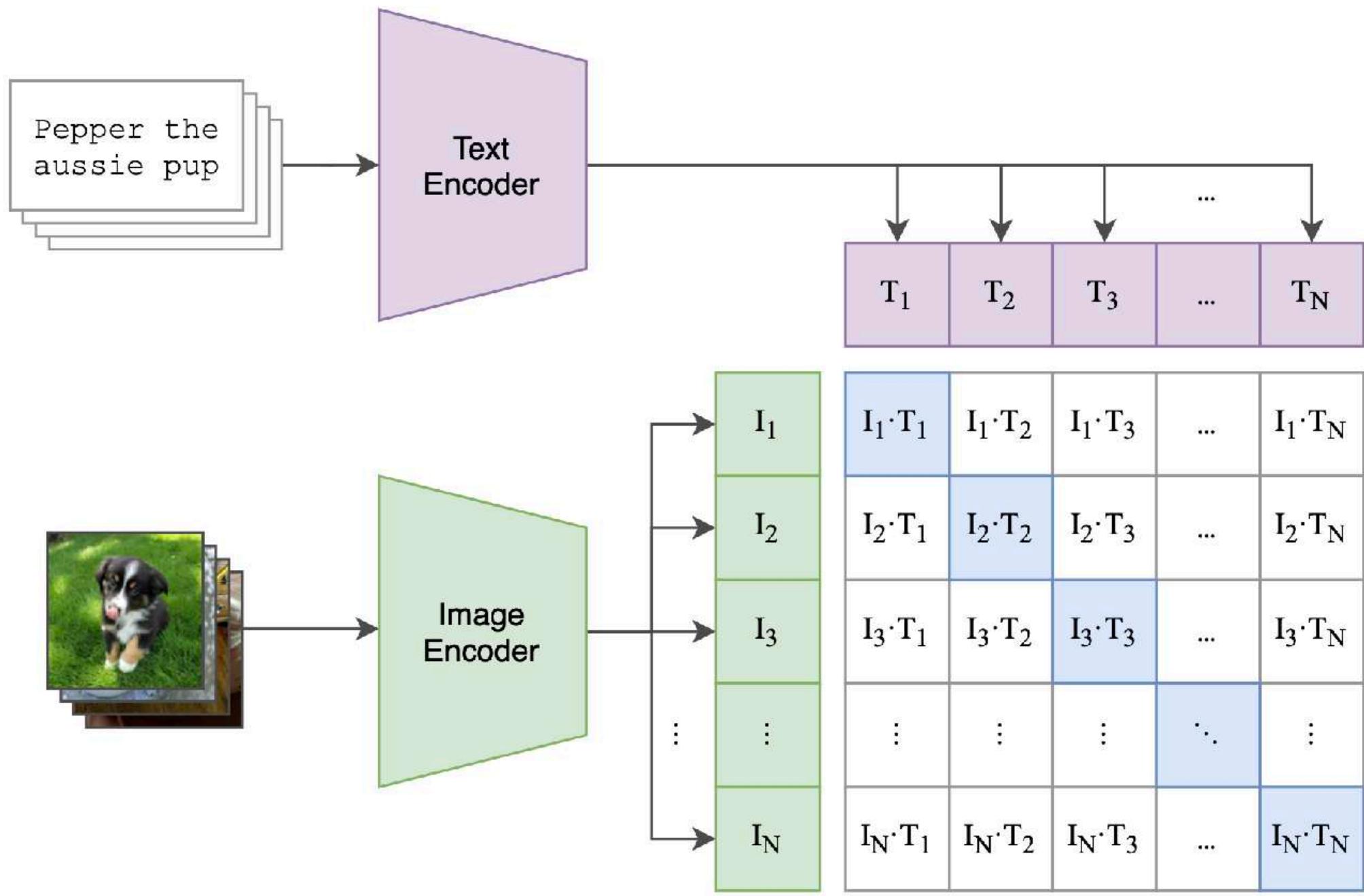
Cross-entropy por columna:

$$\ell_i^{\text{text}} = -\log \frac{\exp(\text{sim}(T_i, I_i)/\tau)}{\sum_{j=1}^N \exp(\text{sim}(T_i, I_j)/\tau)}$$

CLIP

Arquitectura

- Encoder separados para imágenes (ResNet50 o ViT) y textos (Transformer)
- Los tokens del texto son normalizados y linealmente proyectados en el espacio embebido
- Comparten un espacio latente de los embebidos



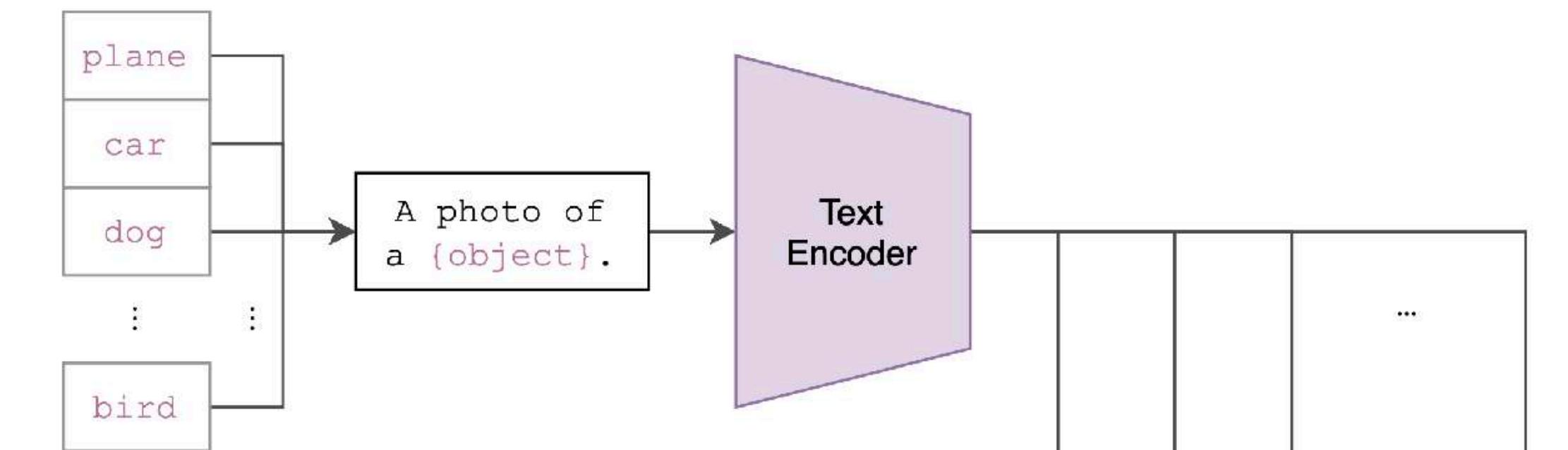
CLIP

Usos comunes

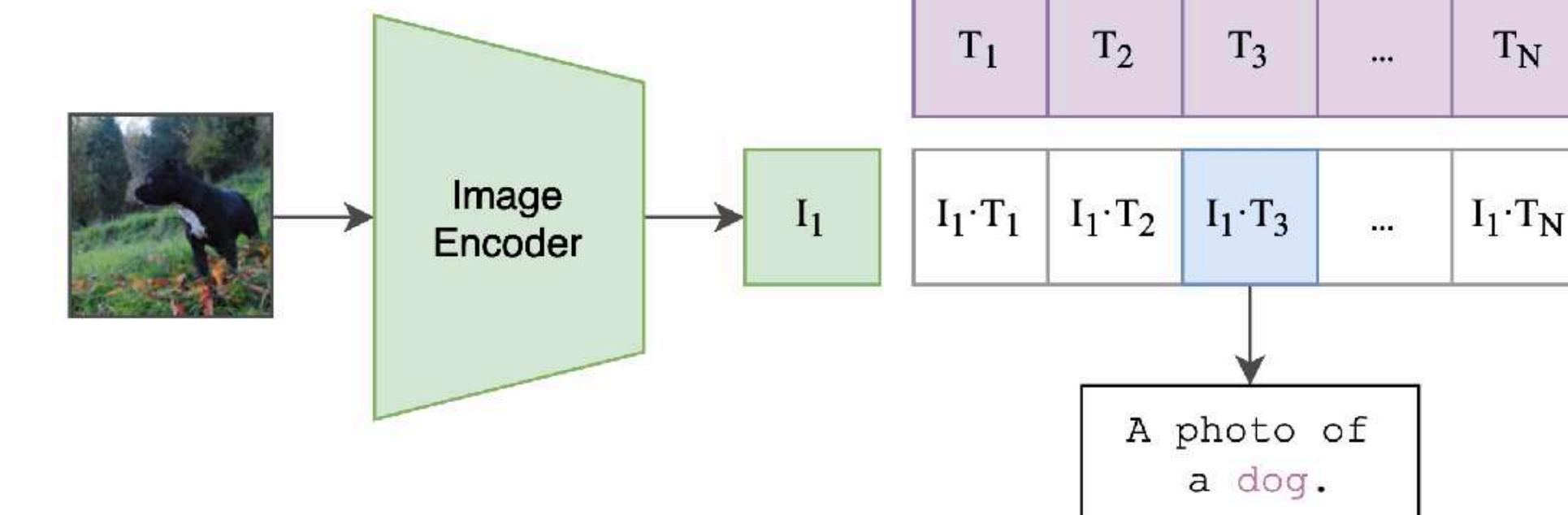
Puede ser usado como:

1. Extractor de características
2. Zero-shot con ingeniería de *prompts*
3. Usar los embebidos para sistemas de *image retrieval*

(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



Agenda



1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
- 3. Transferencia de conocimiento**
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
4. Taller

Transferencia de conocimiento

- Aprovechar los conocimientos aprendidos en una tarea, dominio o modelo (maestro)
- Mejorar el rendimiento o la eficiencia en una tarea, dominio o modelo diferente pero relacionado (estudiante)
- Técnicas como la transferencia de aprendizaje, la adaptación de dominios y la destilación de conocimiento



Agenda



1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
3. Transferencia de conocimiento
 - A. **Transferencia de Aprendizaje**
 - B. Destilación de conocimiento
4. Taller

Transferencia de Aprendizaje

- Tenemos la capacidad para transferir conocimientos entre tareas
- Lo que adquirimos como conocimiento sobre una tarea, lo utilizamos para resolver tareas
- Cuanto más relacionadas estén las tareas, más fácil nos resultará transferir el conocimiento



Transferencia de Aprendizaje

- No aprendemos todo desde cero cuando intentamos aprender nuevos aspectos o temas
- Transferimos y aprovechamos nuestros conocimientos de lo que hemos aprendido en el pasado



Transferencia de Aprendizaje

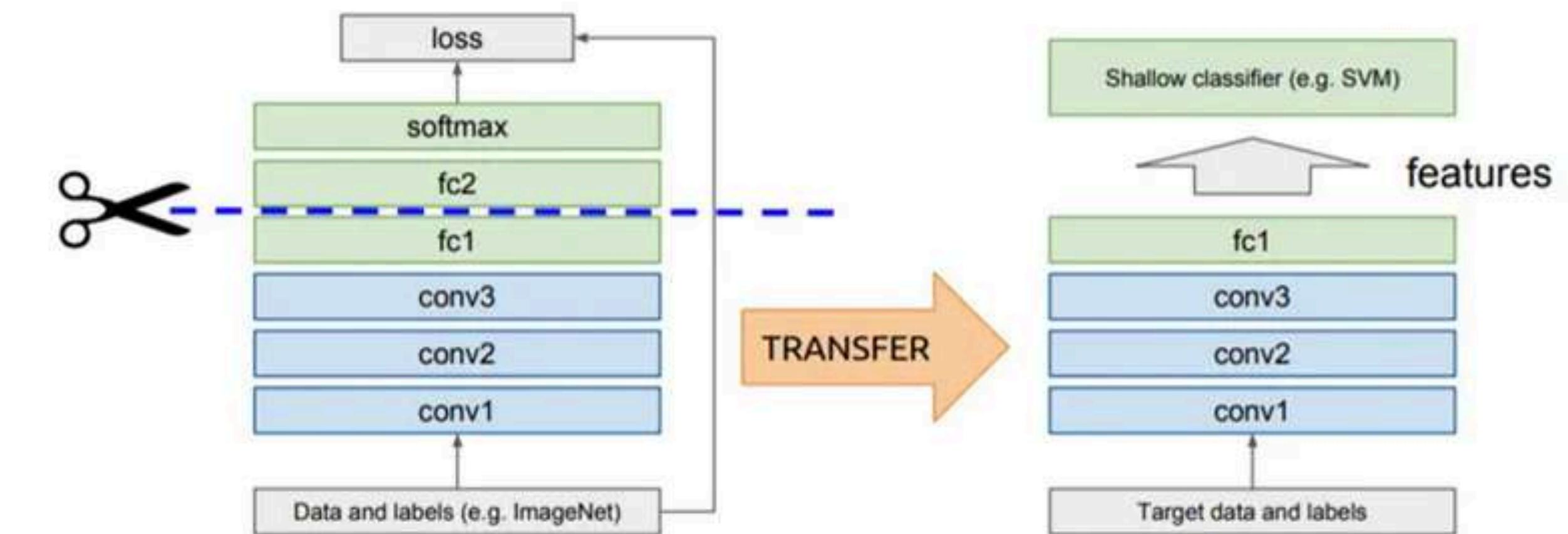
Motivación

- La mayoría de los modelos necesitan una gran cantidad de datos
- Obtener grandes cantidades de datos etiquetados para modelos supervisados es difícil
- Un ejemplo es ImageNet, que contiene millones de imágenes pertenecientes a diferentes categorías



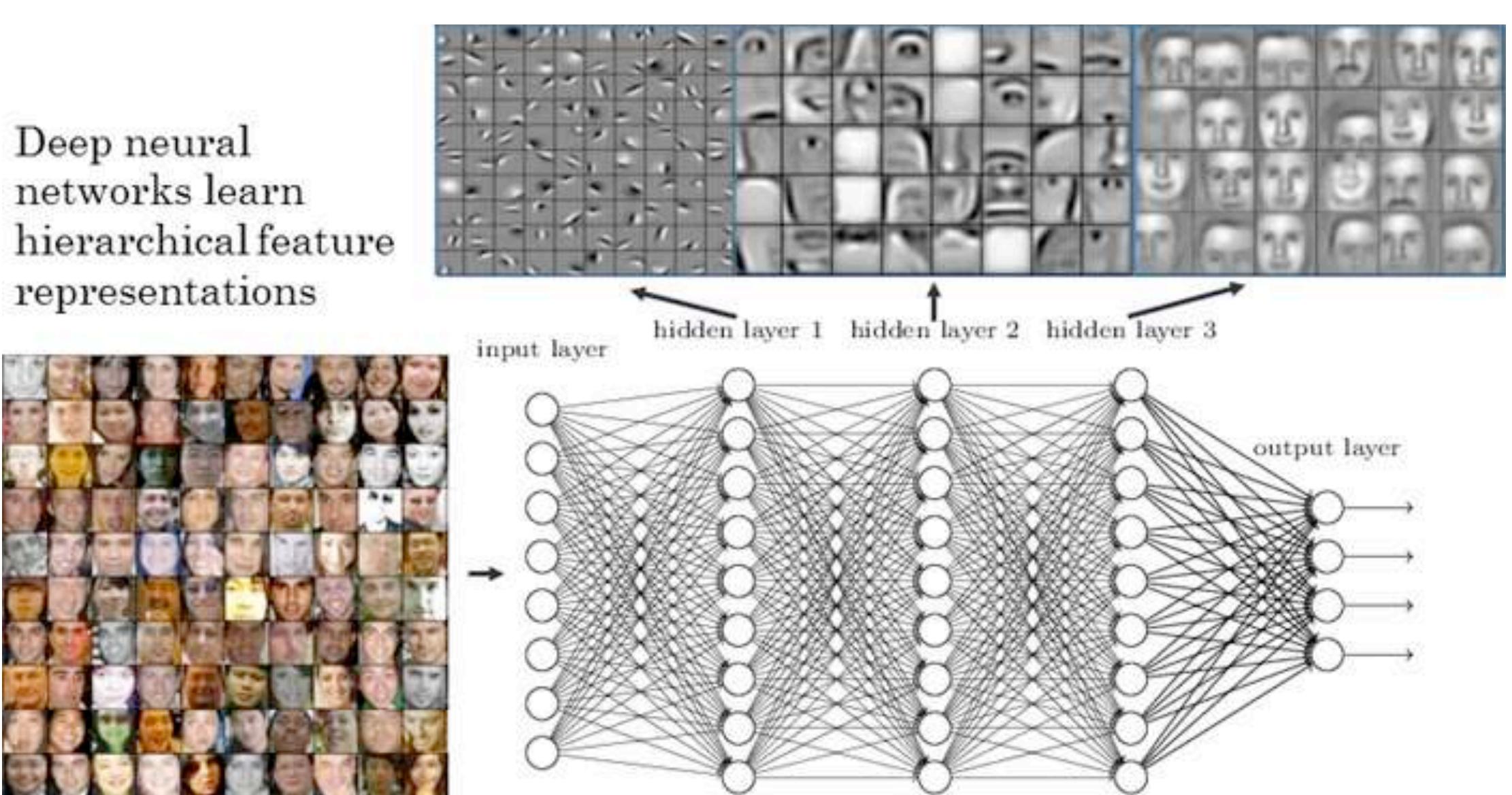
Extractores de Características

- Las arquitecturas de visión por aprendizaje profundo se estructuran en capas
 - a. Extractor de características
 - b. Cabeza de predicción
- Permite utilizar una red pre-entrenada sin sus capas finales como extractor de características

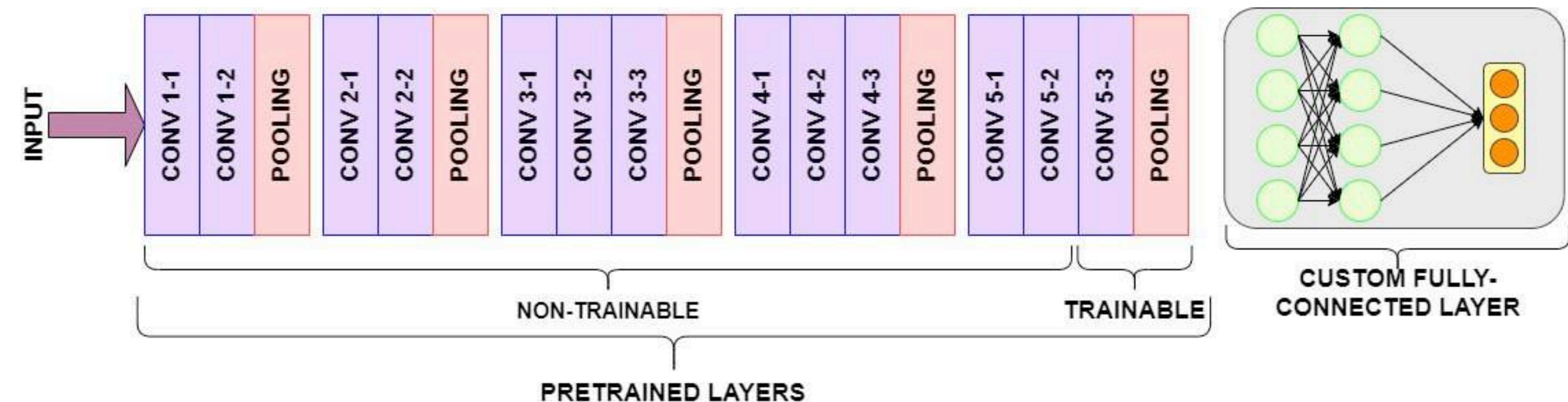


Fine-tuning

- Las capas iniciales captan características genéricas. Las posteriores se centran en la tarea específica
- Congelar (fijar pesos) ciertas capas mientras re-entrenamos y afinar el resto de ellas
- Aprovechamos la arquitectura de la red y utilizamos sus pesos como punto de partida para el re-entrenamiento



“Ajuste suave”



Agenda

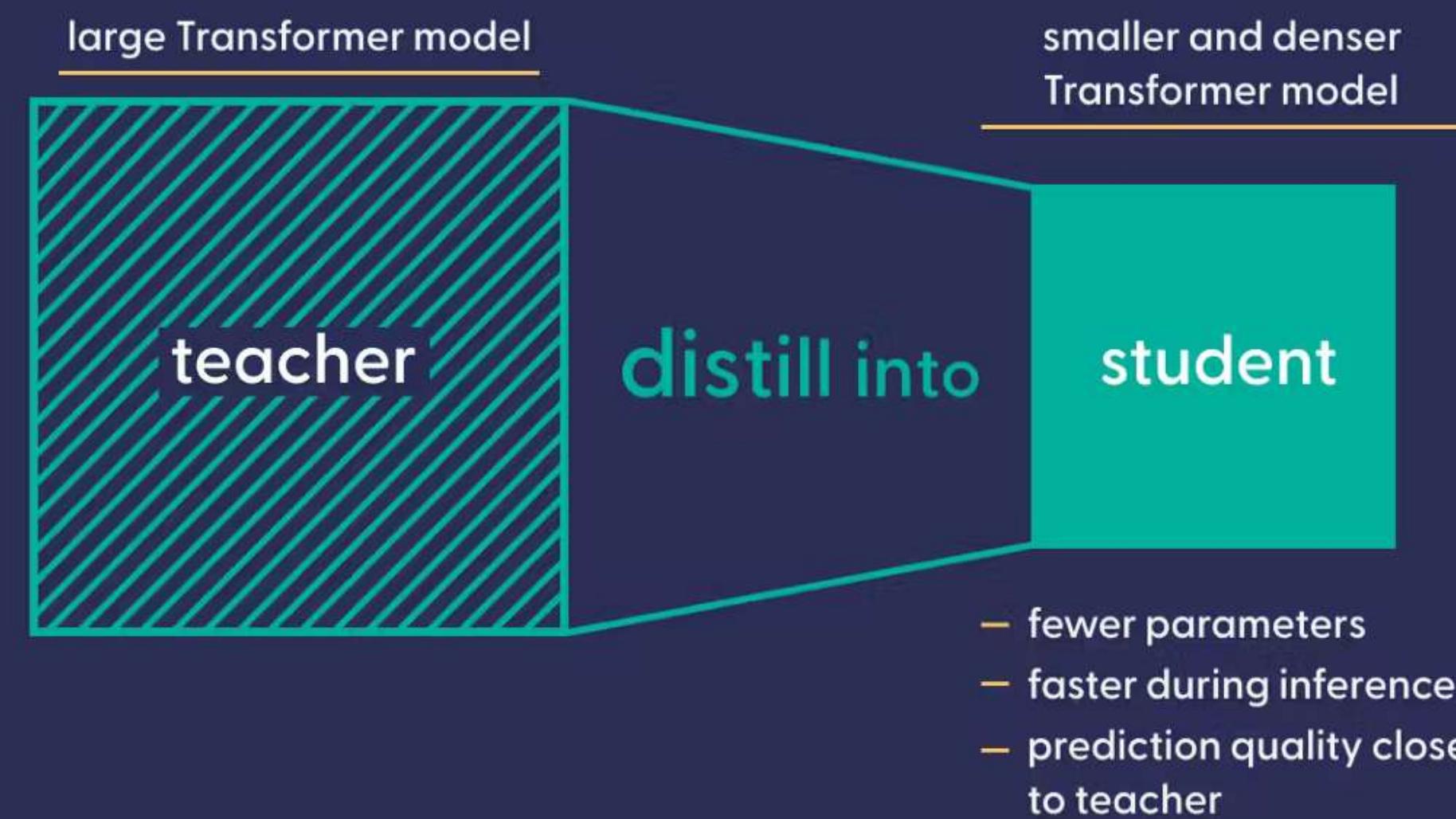


1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento**
4. Taller

KD

Knowledge Distillation

- Objetivo es transferir el aprendizaje de un modelo grande pre-entrenado (modelo maestro) a un (modelo alumno) más pequeño
- Se utiliza como una forma de compresión de modelos y transferencia de conocimientos
- Es normal usar KD para poder usar *Transformers* y *CNN* en producción



Destilación de conocimiento

- Muchos insectos tienen 2 etapas (larvas y adultos) cada una especializada para funciones diferentes
- Pero en ML usamos el mismo modelo para entrenamiento que despliegue teniendo necesidades diferentes:

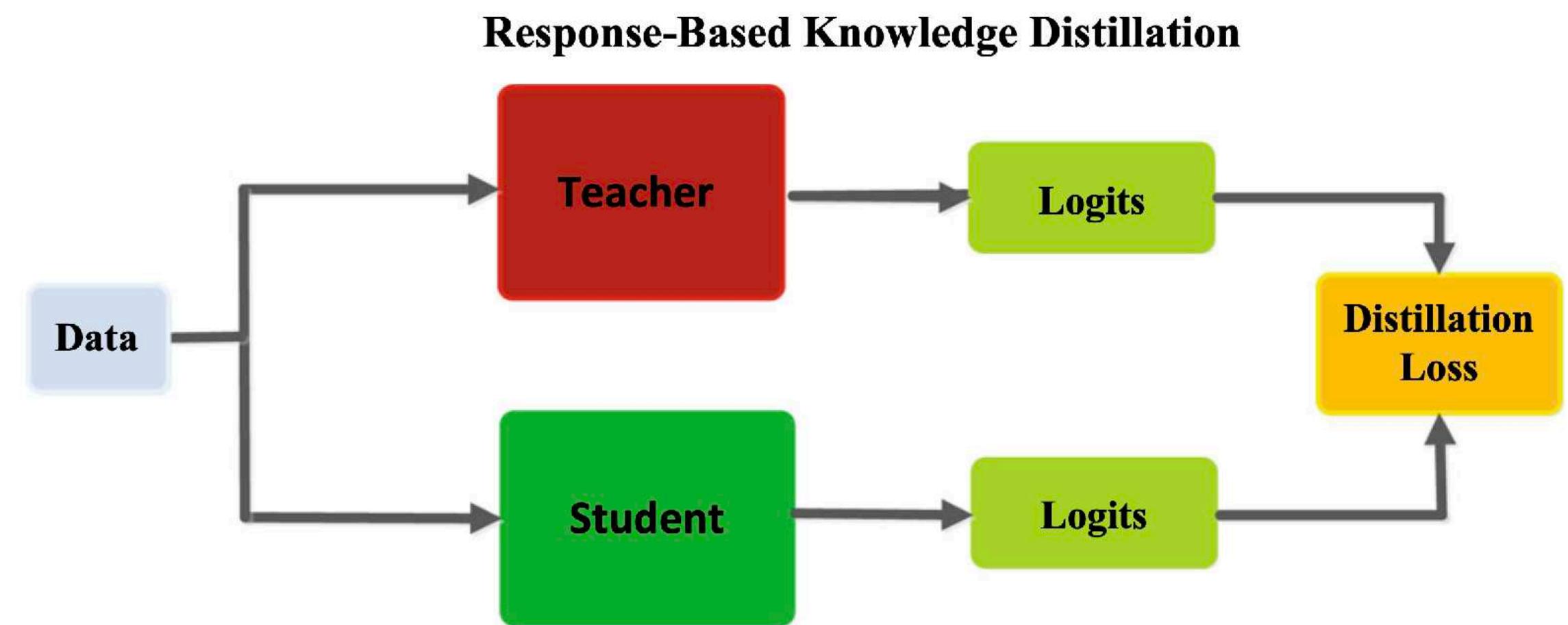
Entrenamiento: modelo poderoso que permite extraer estructura de los datos (ej. ensamble de modelos)

Despliegue: modelo liviano entrenado con un proceso de distillation



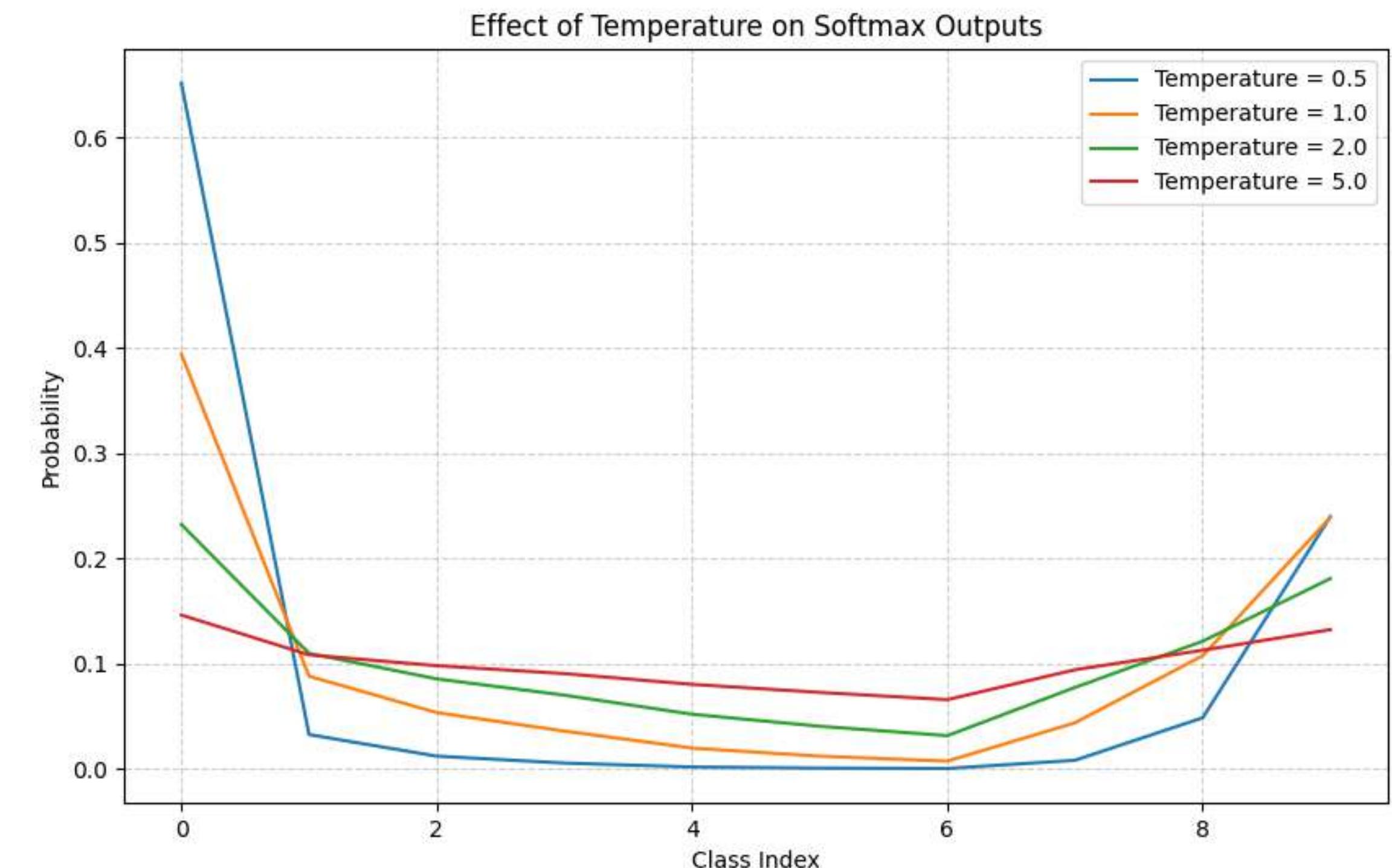
Objetivos Suaves

- Los modelos asignan posibilidades a todas las clases incorrectas inclusive si son muy poco probables (*soft targets*)
- Esta asignación de probabilidades nos dice como el modelo maestro tiende a generalizar
- KD no intenta replicar las salidas del modelo maestro sino emular su forma de pensar (*soft targets* u otras activaciones)



Objetivos Suaves

- Para llegar a las predicciones de un modelo (*hard-targets*) se usan predicciones preliminares con la función *softmax* (*soft-targets*)
- Función *softmax*: convierte los *logits* z_i de cada clase a una probabilidad.
- La temperatura T (normalmente $T = 1$) controla la suavidad de las distribuciones



$$\sigma(z_i) = \frac{e^{(z_i/T)}}{\sum_{j=1}^K e^{(z_j/T)}} \quad \text{for } i = 1, 2, \dots, K$$

Función de Pérdida

Distillation Loss



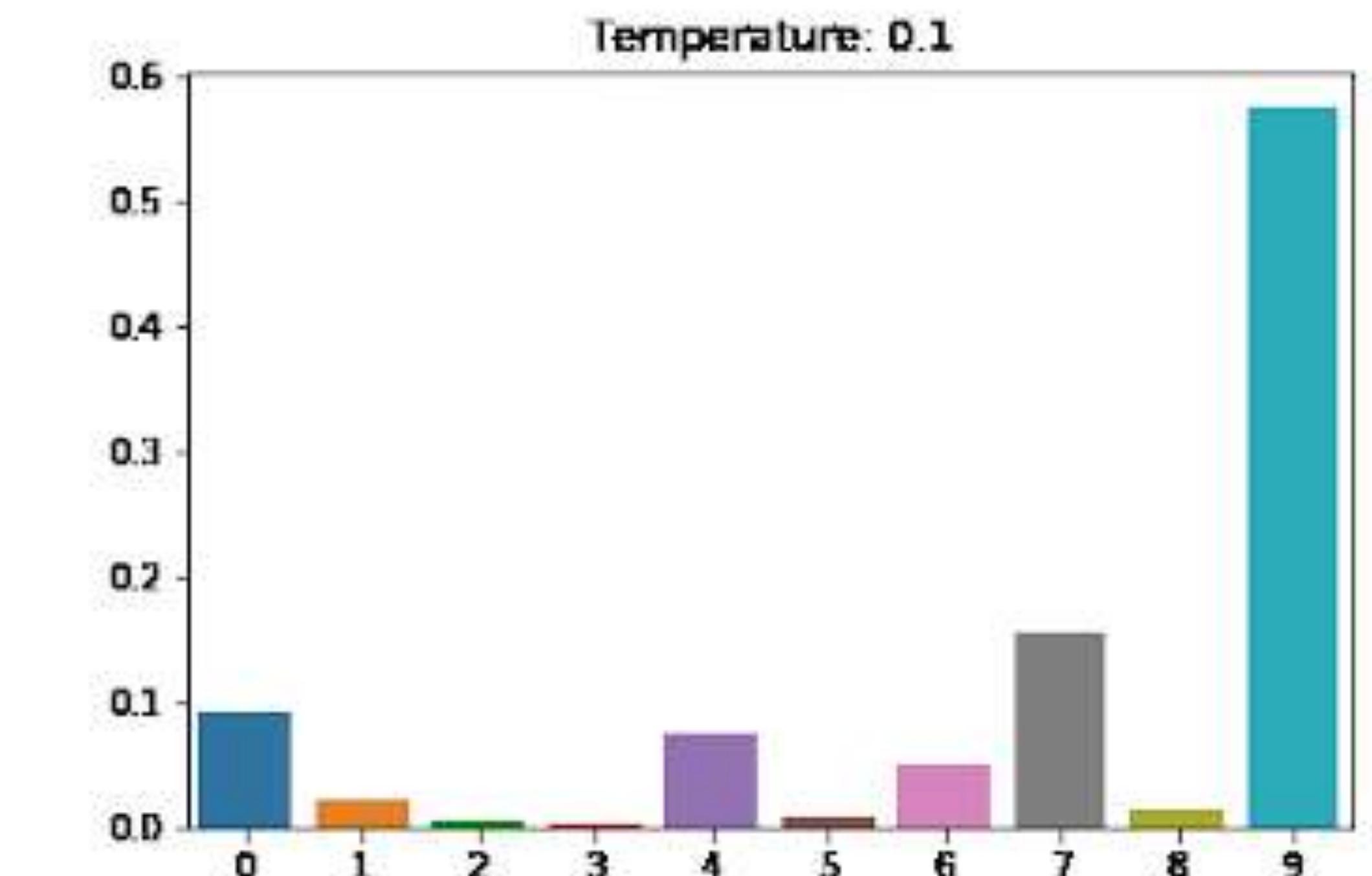
Se emplean dos funciones de pérdida:

Hard-loss: función estándar de la salida del modelo contra las etiquetas

Soft-loss: la diferencia entre las distribuciones de probabilidades de los *soft-targets* (*u otra activación*)

Destilación basada en respuestas

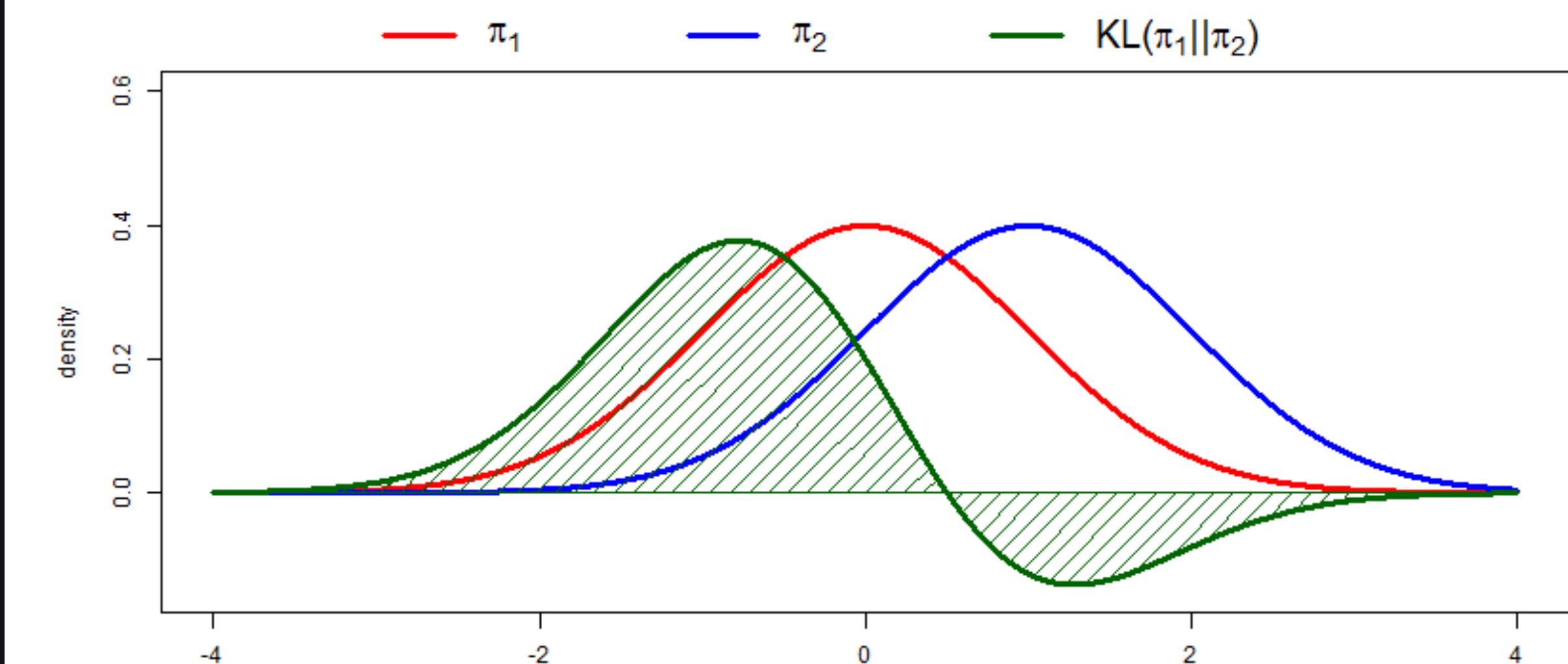
- Se entrena el modelo con *distillation-loss* con última capa del modelo (después de softmax)
- Cuando los modelos están muy seguros tienden a predecir casi un *one-hot vector*
- Se usa valores altos de temperatura para aumentar la entropía de las respuestas del modelo maestro
- Esta aproximación falla para supervisar las representaciones intermedias



Función de perdida suave

- *Kullback-Leibler divergence (KL divergence)* es usada para medir que tan diferentes son las dos distribuciones
- La función de pérdida utiliza la divergencia discreta KL entre las salidas softmax del modelo maestro $p(x)$ y del modelo estudiante $Q(x)$ para cada clase x

$$D_{\text{KL}}(p \parallel q) = \sum_{x \in \mathcal{C}} p(x) \log \frac{p(x)}{q(x)}$$



$$D_{\text{KL}}(P \parallel Q) = \int P(x) \ln \left(\frac{P(x)}{Q(x)} \right) dx$$



Destilación de Conocimiento

Hay 3 formas de KL:

1. Basado en respuestas
2. Basado en características
3. Basado en relaciones

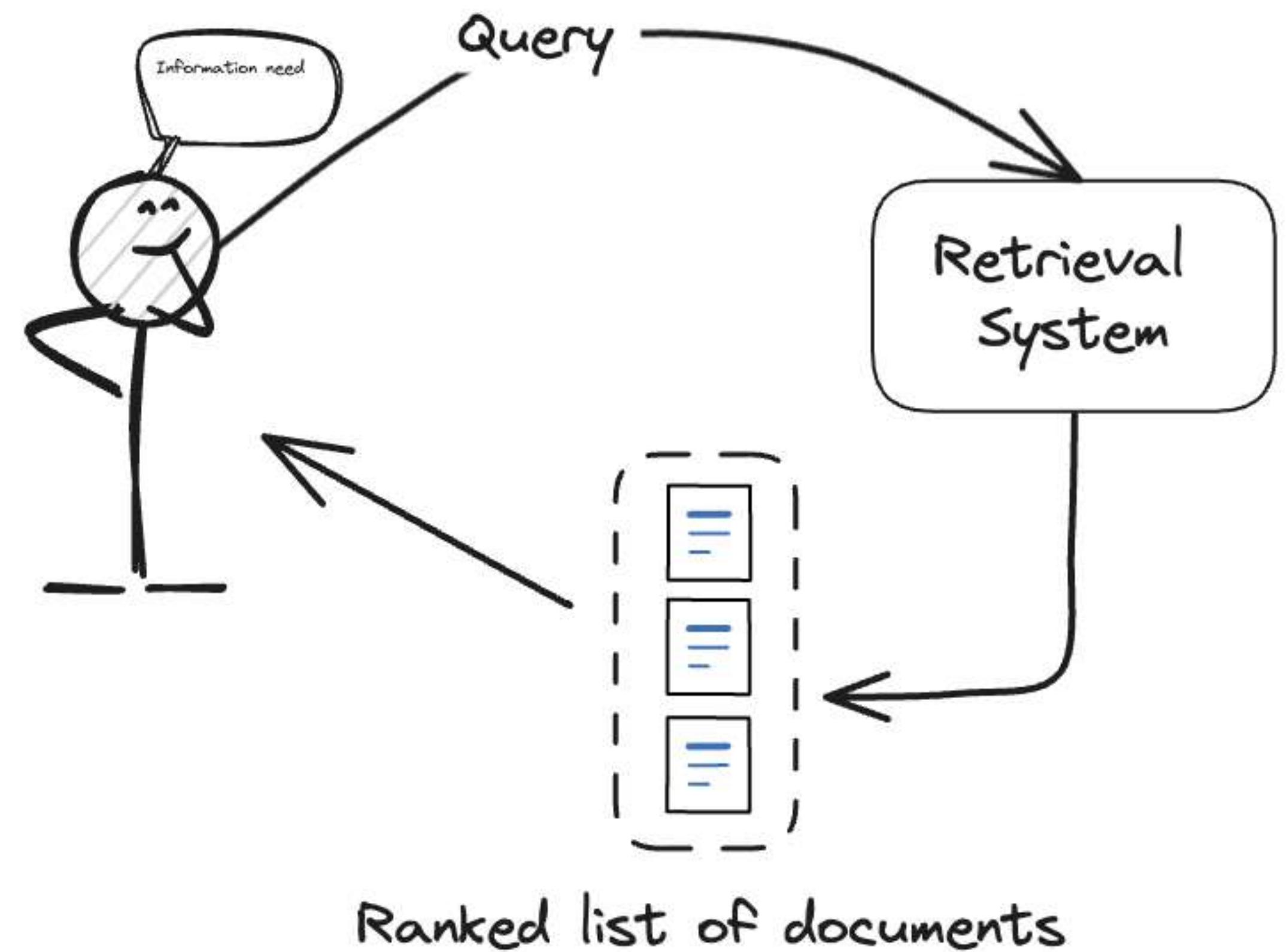
Agenda



1. CNN sesgo inductivo
2. Vision Image Transformers (ViT)
3. Transferencia de conocimiento
 - A. Transferencia de Aprendizaje
 - B. Destilación de conocimiento
- 4. Taller**

Taller

- Implementar un sistema de recuperación texto-imagen (*image retrieval*)
- Usar imágenes de Caltech 256 (tomar 20%-50% imágenes aleatorias)
- Link dataset



Lecturas

2022 - A ConvNet for the 2020s

2021 - Training data efficient image transformers & distillation through attention

Referencias

2015 - Distilling the Knowledge in a Neural Network

2017 - Attention is all you need

2021 - Knowledge Distillation: A Survey

2021 - An Image Is Worth 16X16 Words Transformers For Image Recognition At Scale

2021 - Swin Transformer Hierarchical Vision Transformer using Shifted Windows

2021 - Knowledge Distillation A Survey

2021 - A Comprehensive Survey on Transfer Learning

2021 - Learning Transferable Visual Models From Natural Language Supervision