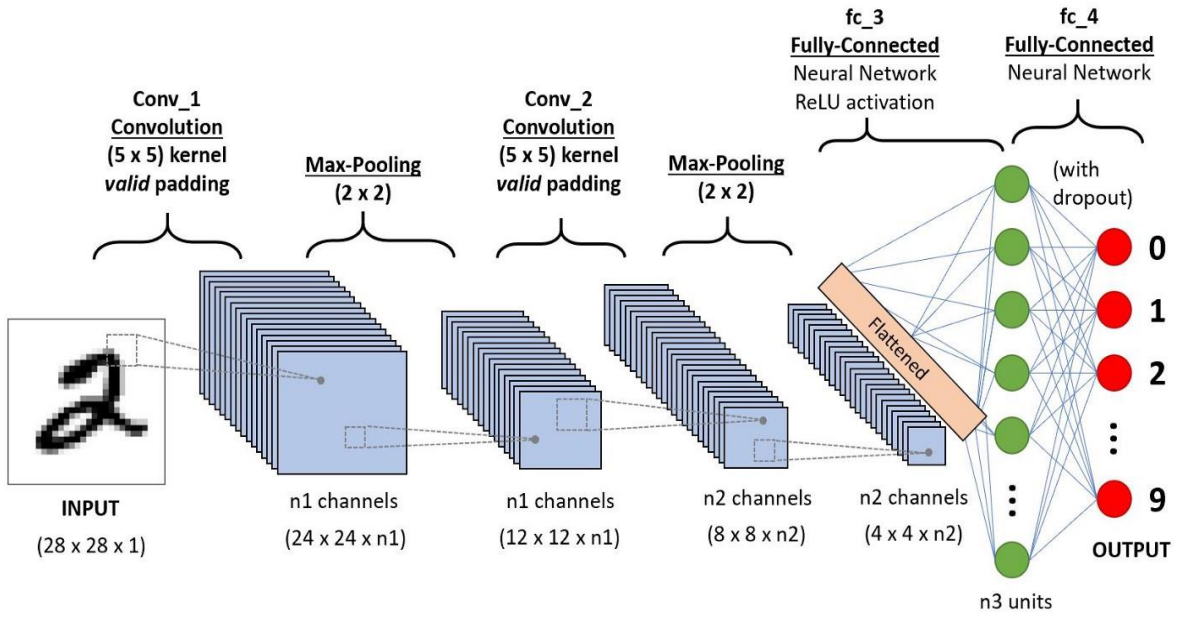
Visión computacional moderna y modelos multimodales

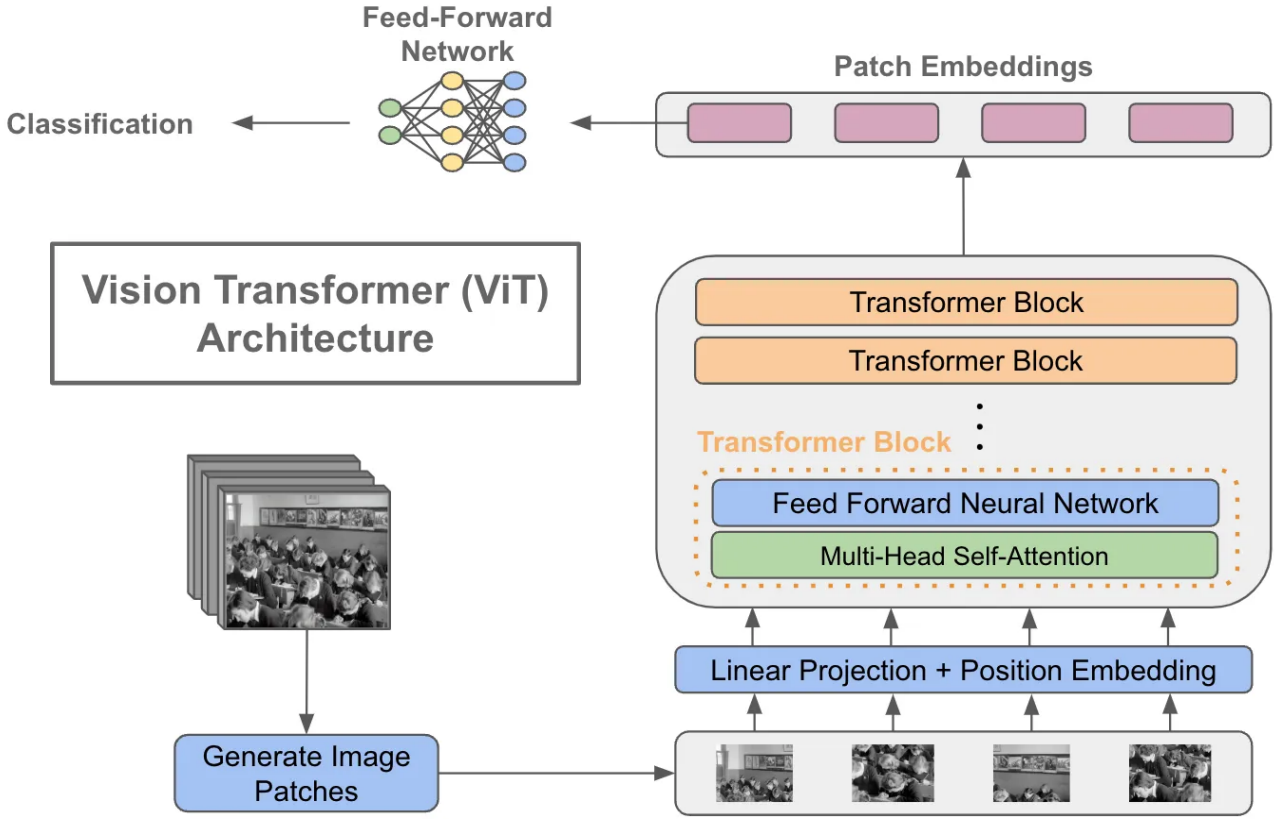
De ViT -> CLIP -> VQA -> MLLM (Q-Former/LLM)

Temario

- De CNNs a Vision Transformers (ViT): motivación, parches y encoder.
- Representaciones conjuntas texto-imagen: CLIP y aprendizaje contrastivo.
- Recuperación multimodal: text<->image, matriz de similitud y métricas (Recall@K).
- Modelos visión-lenguaje: captioning, grounding y VQA (Visual Question Answering).
- LLM multimodales (MLLM): ViT/encoders + adaptadores (Q-Former) + LLM; audio y vídeo.
- Aplicaciones: salud, industria, educación y sistemas interactivos.

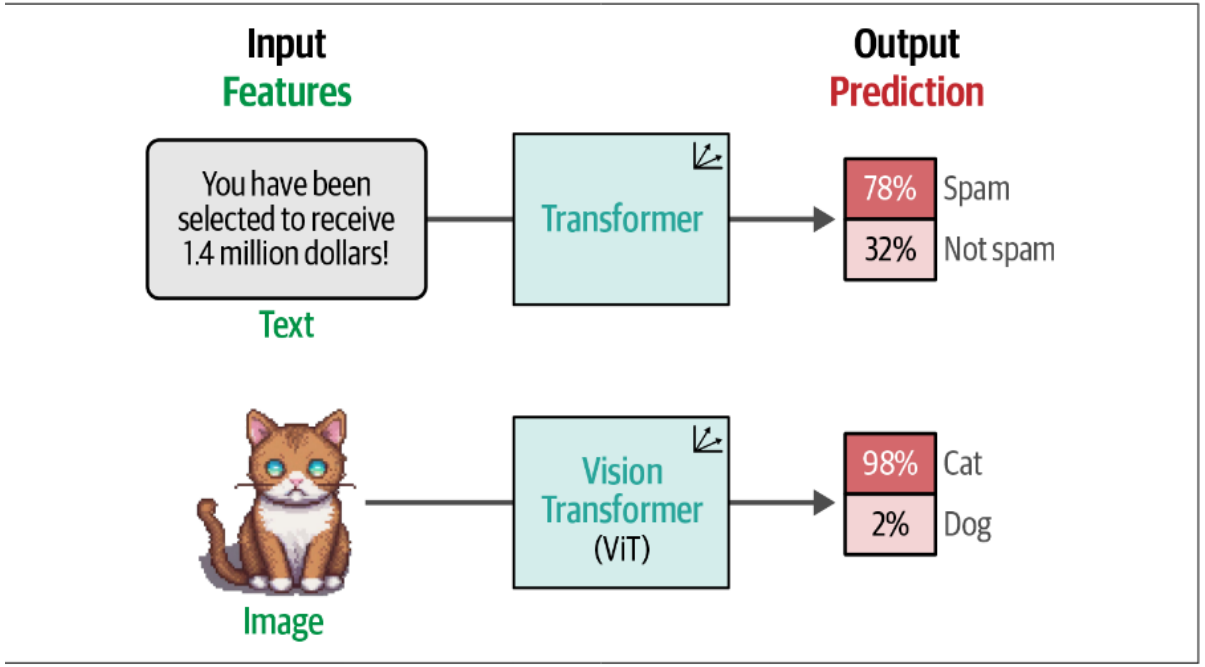
De convoluciones a Vision Transformers (ViT)

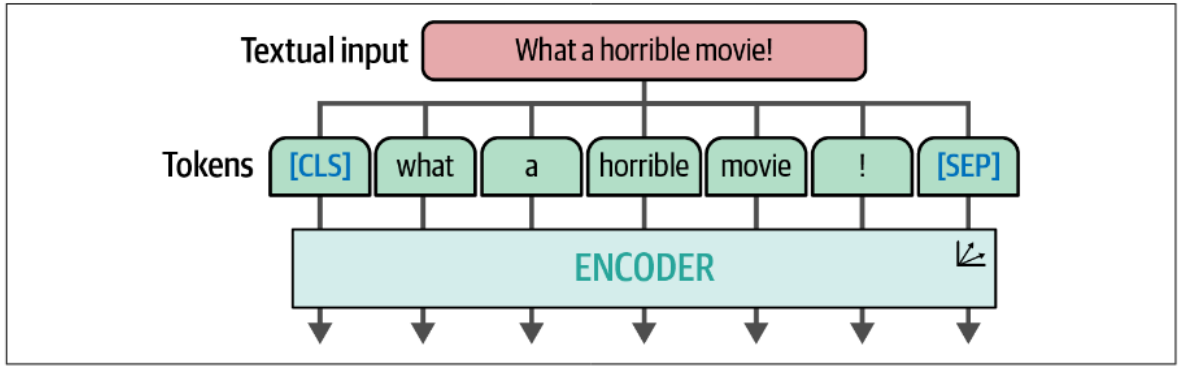




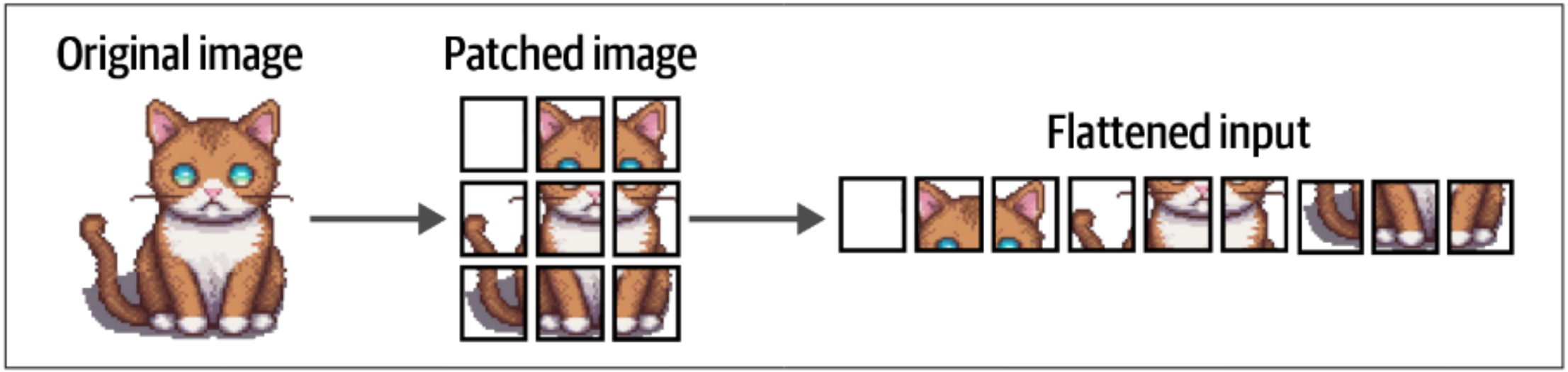
De CNNs a ViT: ¿por qué cambiar?

- CNNs: fuerte sesgo inductivo (localidad y traslación) -> muy eficientes en datos "pequeños".
- Transformers: escalan muy bien con datos y cómputo; se benefician de preentrenamiento masivo.
- ViT trata una imagen como una secuencia de tokens (parches) y aplica el mismo "motor" (self-attention).
- Resultado: arquitectura más uniforme para visión y lenguaje; facilita modelos multimodales.



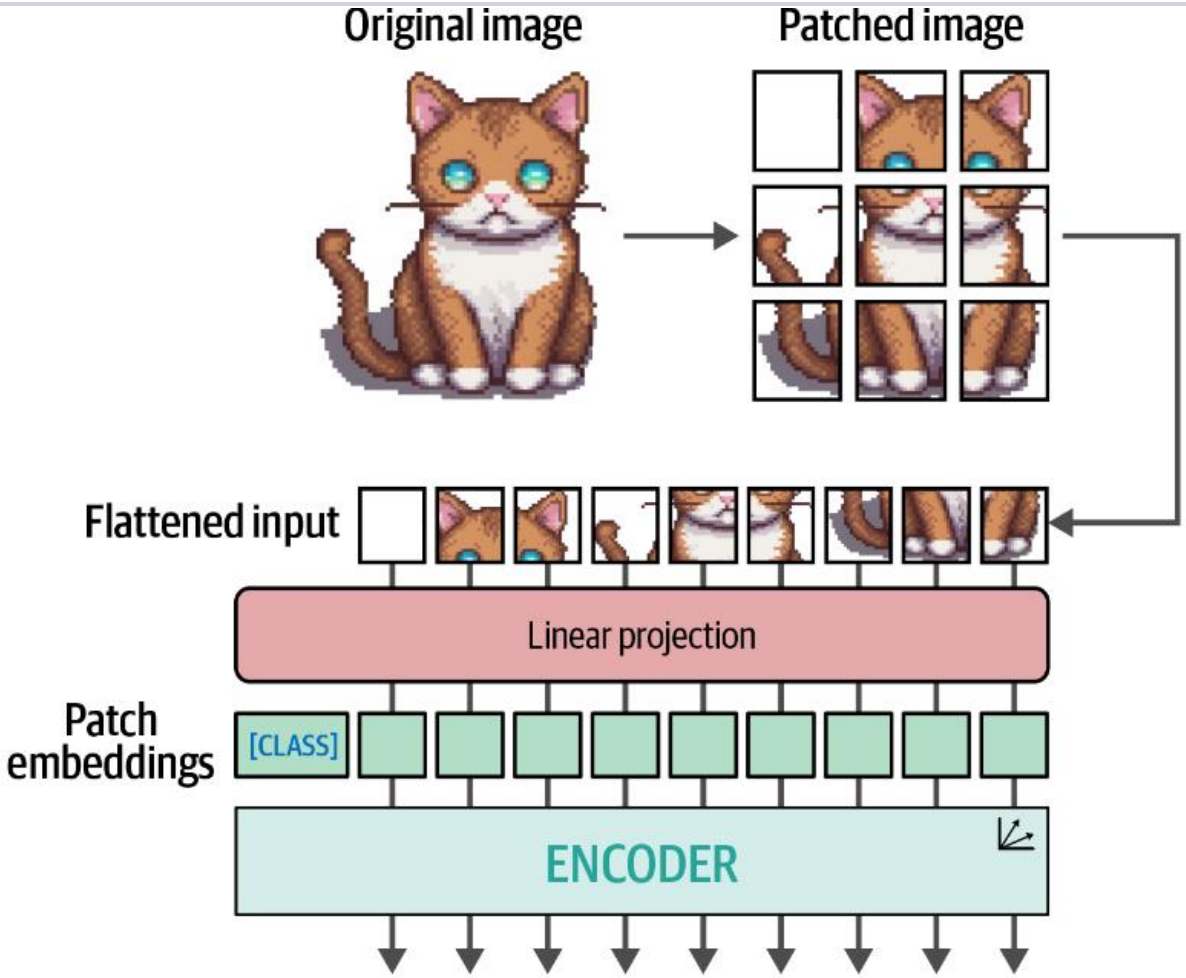


ViT: de píxeles a tokens



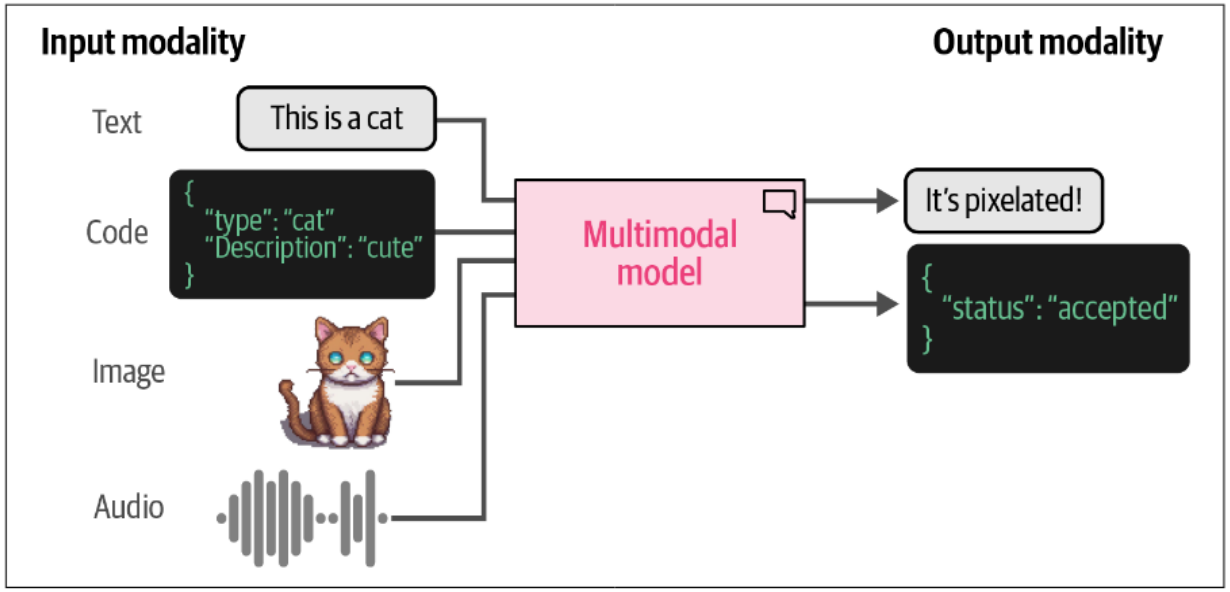
- Dividir la imagen en parches fijos (por ejemplo, 16×16).
- Aplanar cada parche y proyectarlo a un embedding (vector) con una capa lineal.
- Añadir embeddings posicionales para preservar el orden espacial.
- Procesar la secuencia con un encoder Transformer estándar.

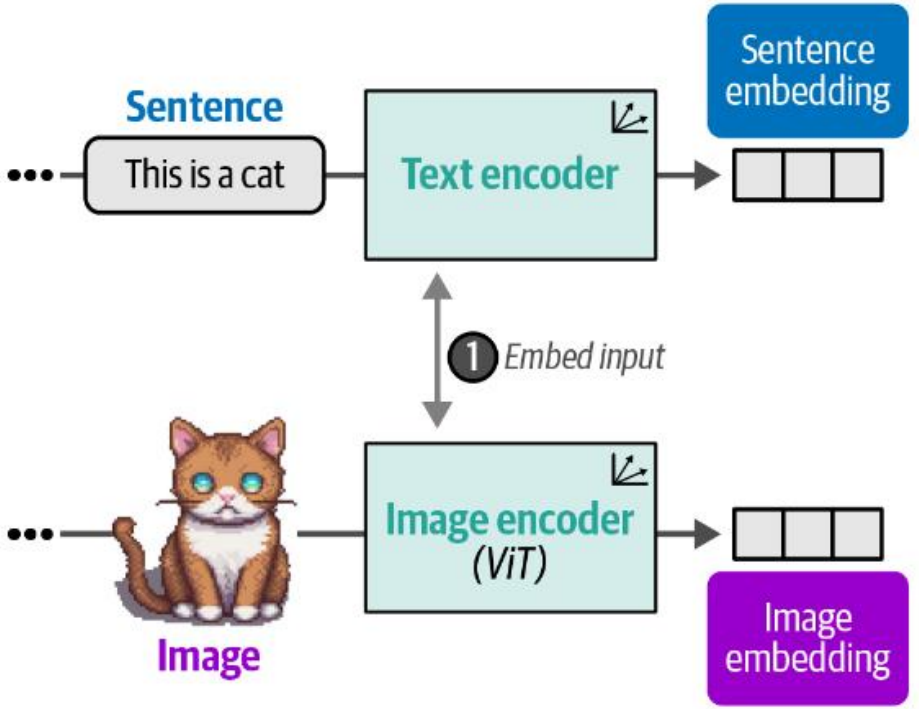
Algoritmo del Vision Transformer (ViT)



Token [CLASS] resume la imagen para tareas de clasificación; también es común "pooling" o tokens extra.

Modelo de embeddings multimodal

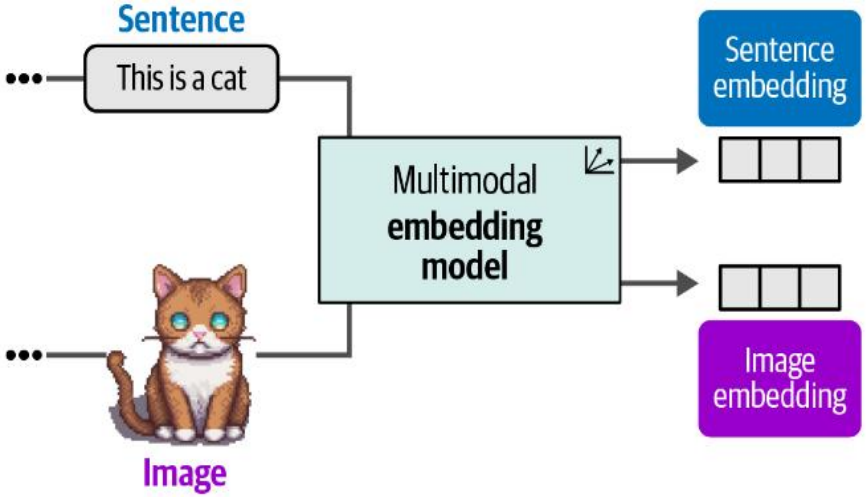


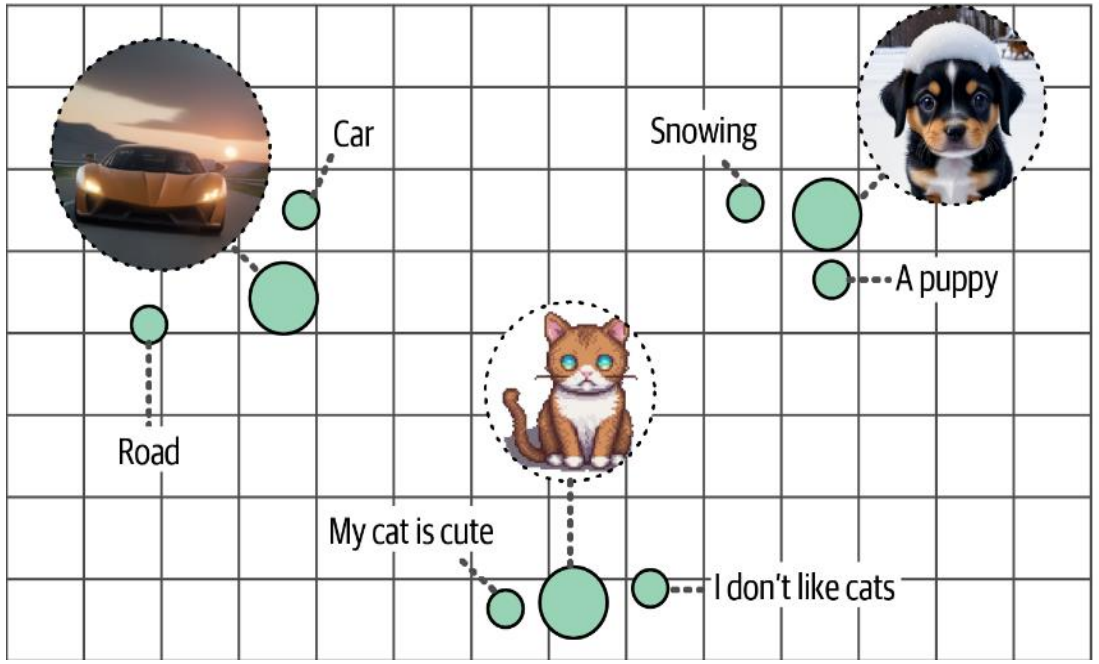


Los modelos de embeddings multimodal pueden crear embeddings de múltiples modalidades en el mismo espacio vectorial.

Representaciones conjuntas texto-imagen

- Objetivo: mapear imagen y texto a embeddings comparables (por ejemplo, similitud coseno).
- Si están "cerca" -> son semánticamente compatibles. Esto permite recuperación: texto->imagen y imagen->texto.
- También permite clasificación "zero-shot" usando prompts.

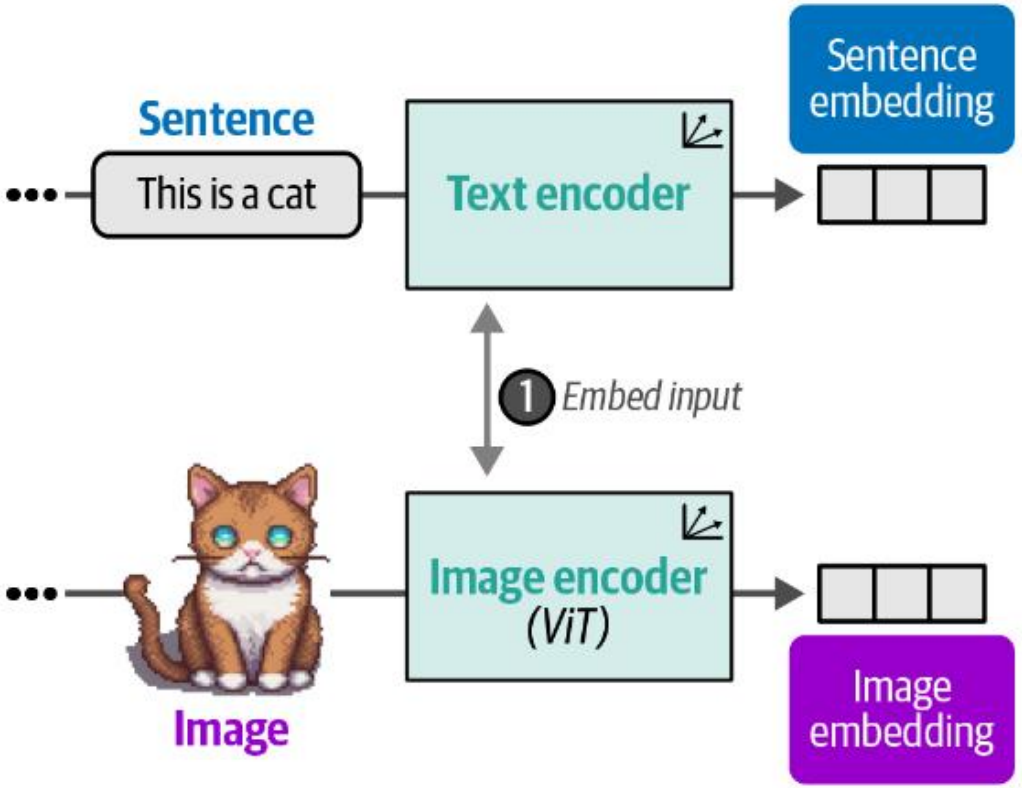




Tipos de datos para entrenar un modelo de embedding

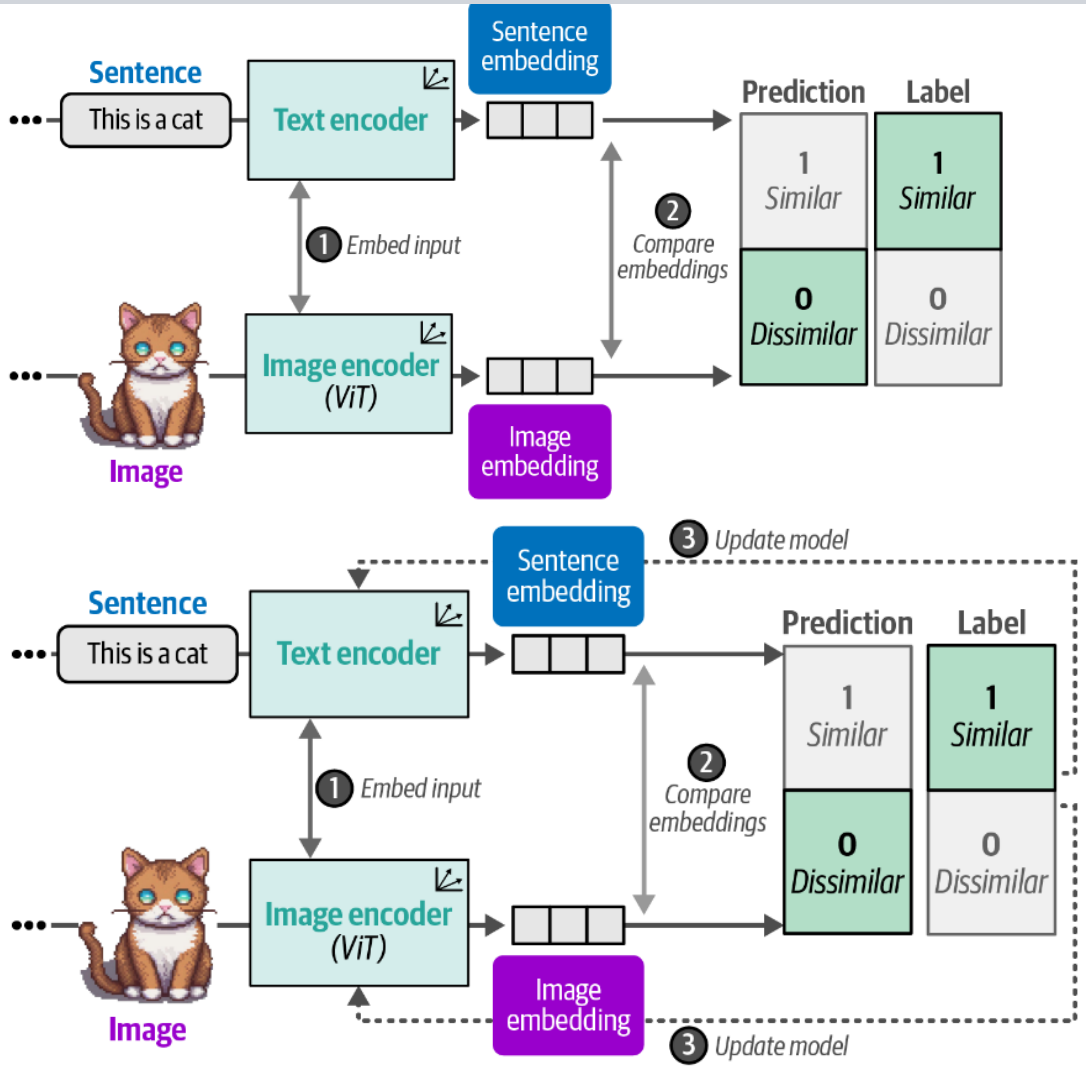
- Cada ejemplo alinea una imagen con su descripción (caption).
- La diversidad de datos mejora robustez y generalización.
- La tarea contrastiva aprende a distinguir el caption correcto de captions "negativos" en el batch.

Entrenamiento de CLIP : primer paso



En el primer paso del entrenamiento de CLIP, se incorporan imágenes y texto utilizando un codificador de imágenes y texto, respectivamente.

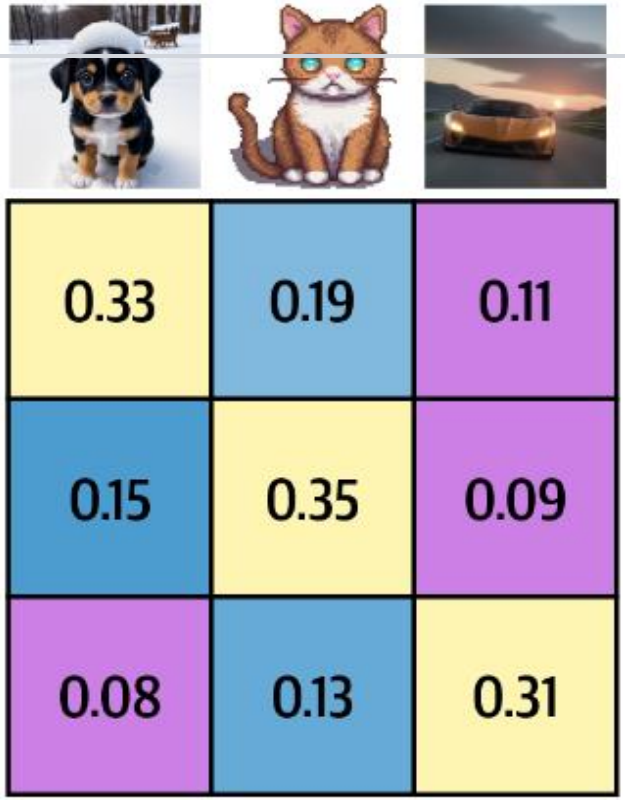
Entrenamiento de CLIP: comparación y actualización



Se calcula la similitud entre la oración y la imagen con embedding mediante la similitud del coseno.

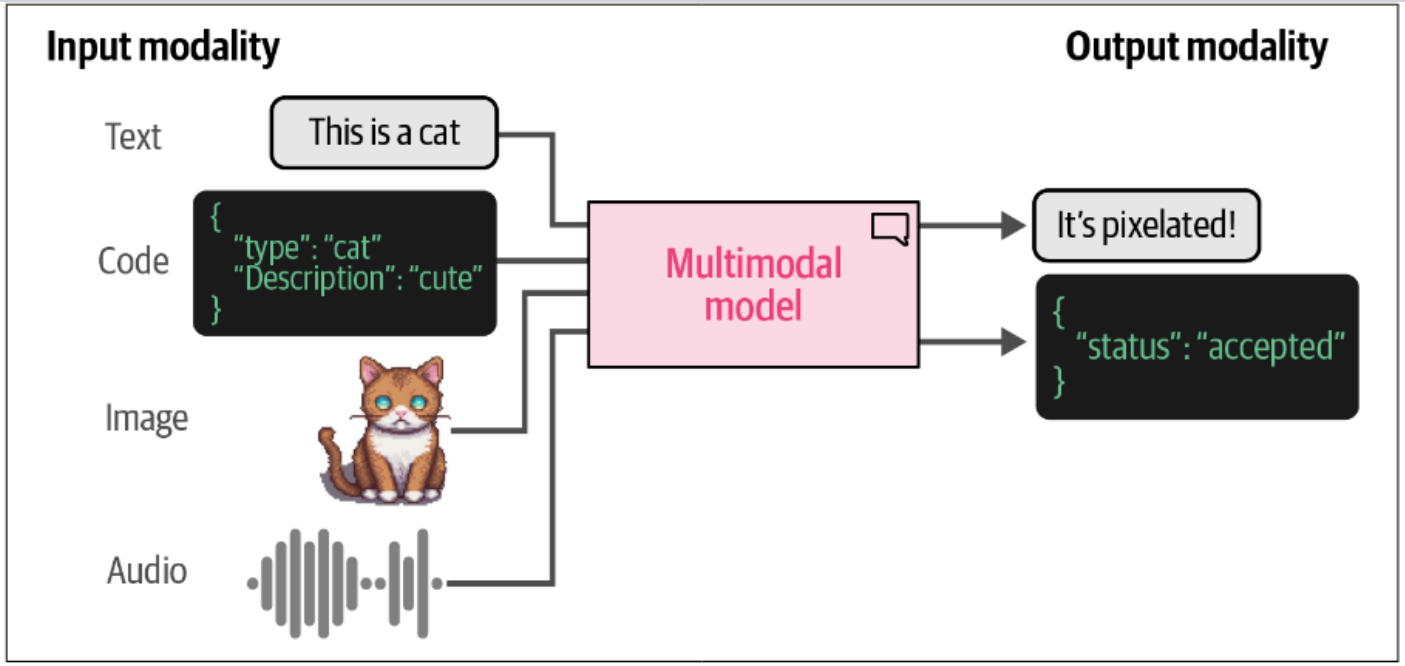
Se actualizan los codificadores de texto e imagen para que coincidan con la similitud deseada. Esto actualiza los embedding para que estén más próximas en el espacio vectorial si las entradas son similares.

Matriz de similitud CLIP



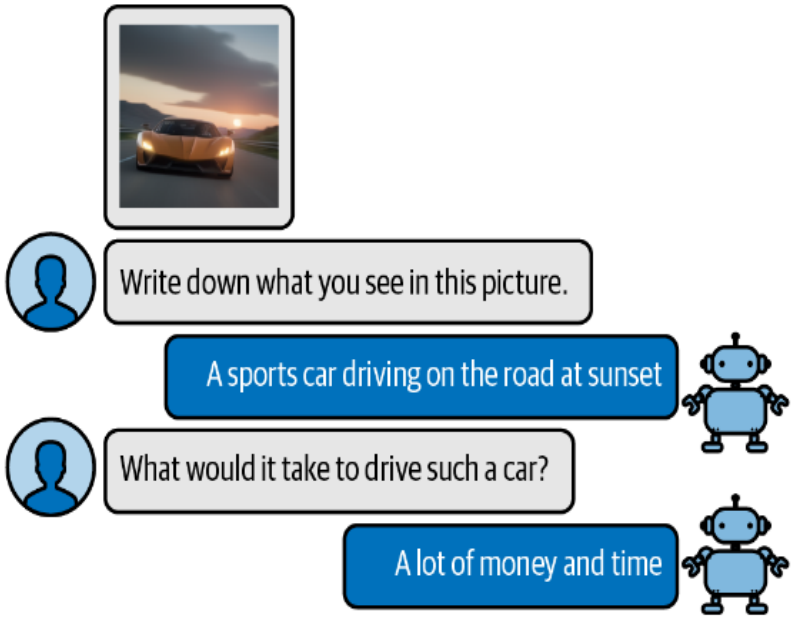
- Recuperación texto->imagen: rankear imágenes por similitud con el texto.
- Recuperación imagen->texto: rankear captions por similitud con la imagen.
- Métrica típica: Recall@K, mAP; también "prompt ensembling" para clasificación.

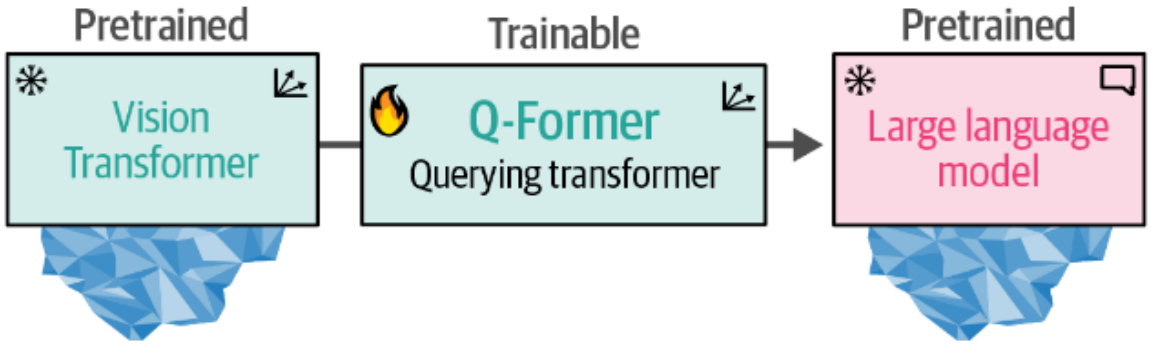
Patrón moderno: "prompting" multimodal



- Un modelo puede: describir, responder, extraer información (OCR), razonar con contexto visual.
- Se apoya en representaciones preentrenadas (visión) + un LLM para lenguaje e instrucciones.
- El desafío es alinear: qué ve el encoder vs qué "dice" el LLM.

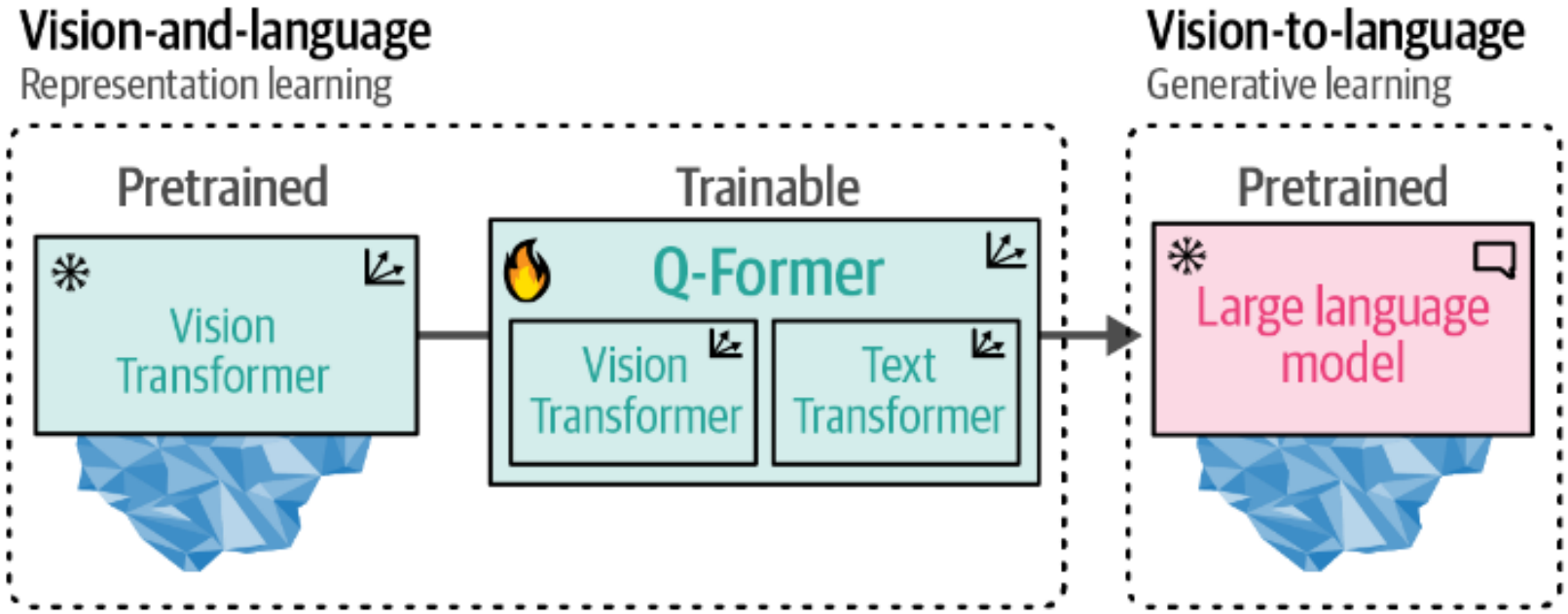
Arquitectura visión-lenguaje para VQA y MLLM





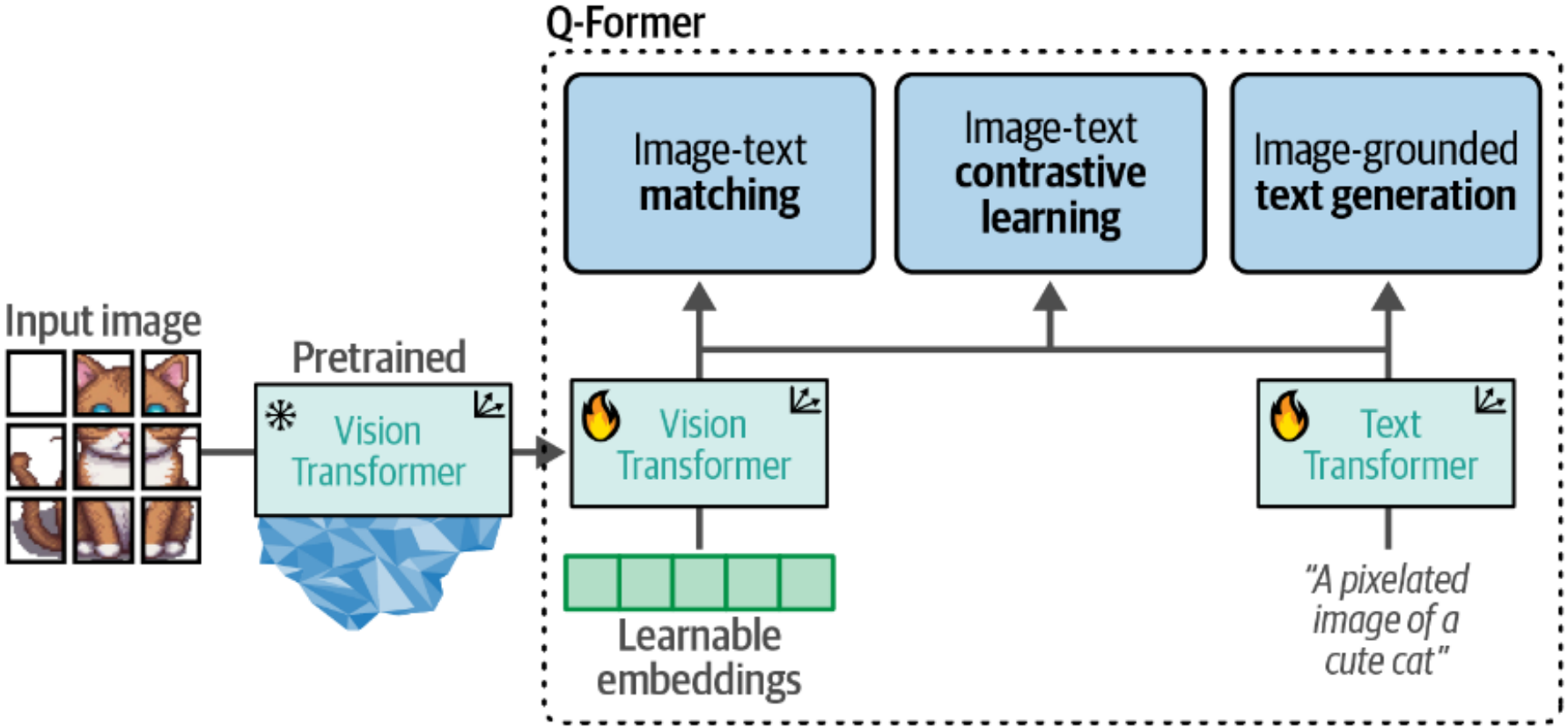
- Congelar módulos grandes (ViT/LLM) reduce costo de entrenamiento.
- El adaptador aprende a traducir "features visuales" a tokens/embeddings consumibles por el LLM.
- Ventaja: reutiliza capacidades del LLM (instrucciones, planificación, lenguaje).
- Extensión natural: audio/video con encoders dedicados (por ejemplo, wav2vec/AST; TimeSformer, etc.).

BLIP-2 /Q-Former: dos etapas



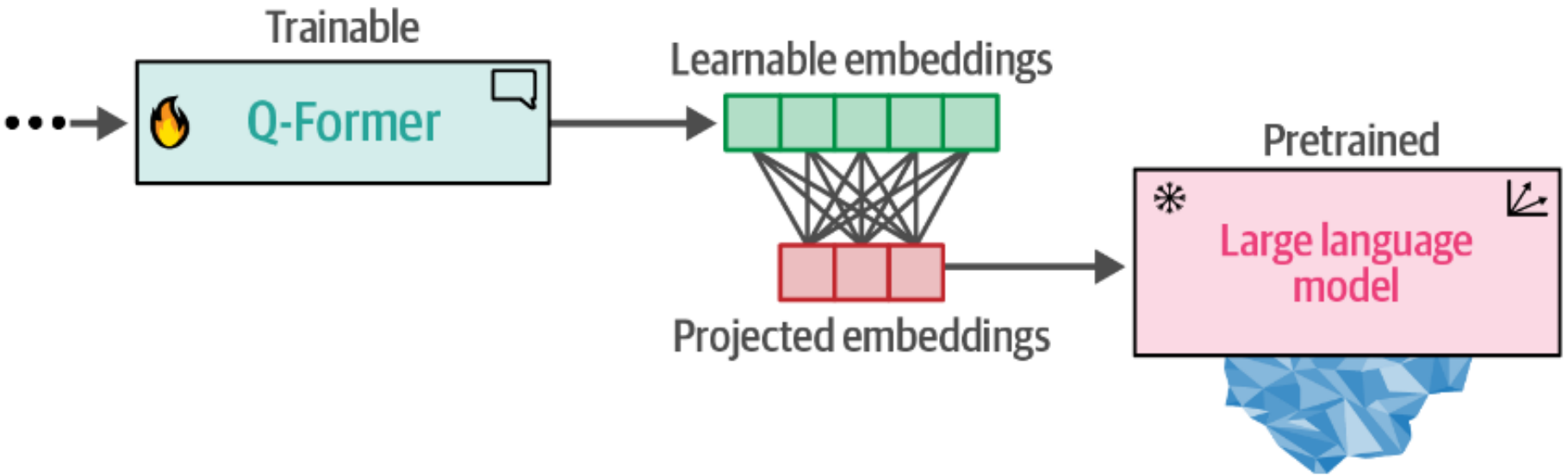
- Etapa 1: aprender consultas (queries) que extraen información relevante del encoder visual.
- Etapa 2: proyectar esas consultas al "espacio" del LLM para generar texto condicionado en la imagen.

Primer paso de BLIP-2



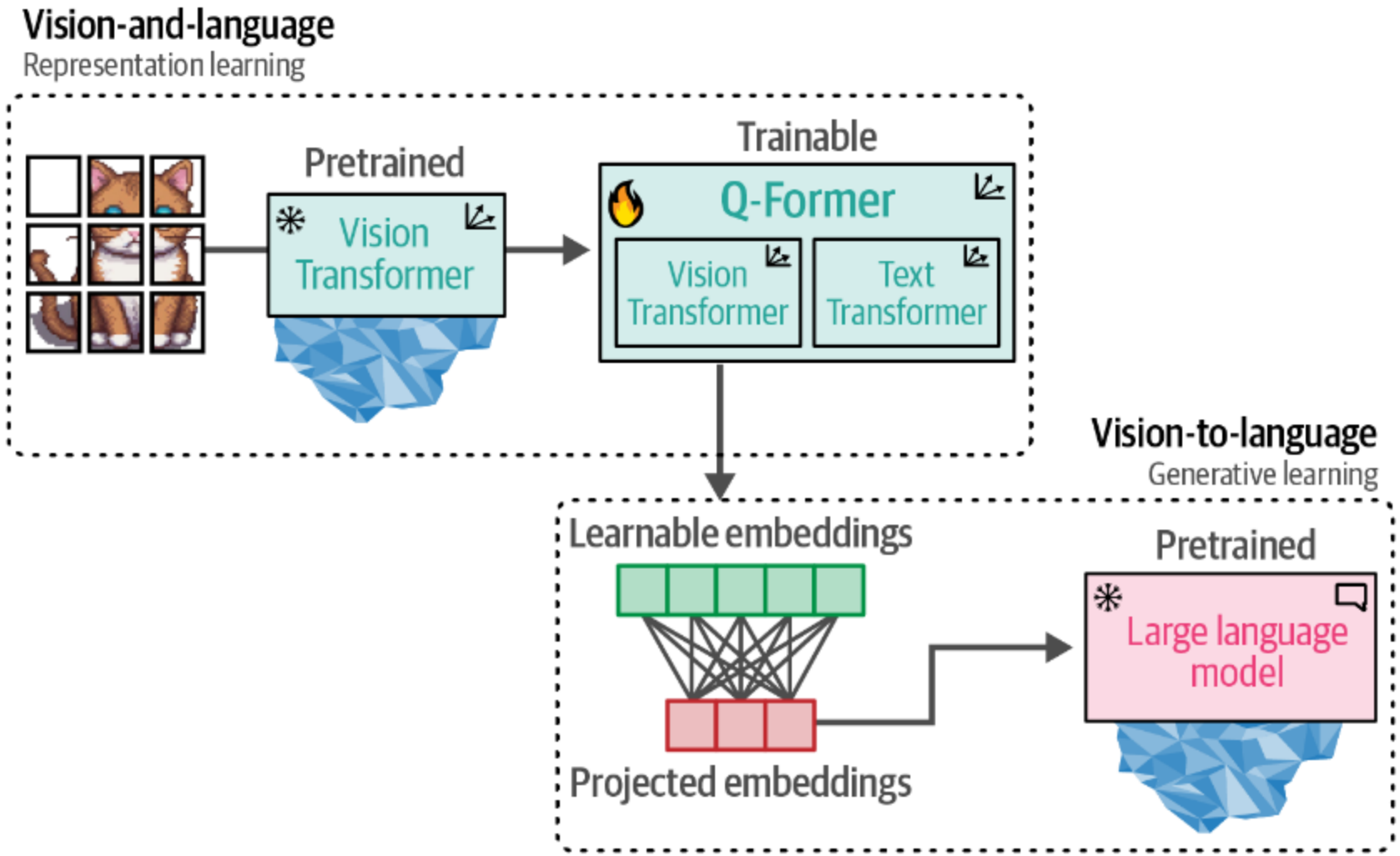
- Image-text matching: ¿corresponden?
- Contrastive learning: organizar embeddings para retrieval.
- Image-grounded generation: generar texto "anclado" en la imagen.

Segundo paso de BLIP-2



Los embeddings aprendidos del Q-Former se transfieren al LLM mediante una capa de proyección.
Las embeddings proyectados sirven como un suave recordatorio visual.

El procedimiento completo de BLIP-2



Aplicaciones: salud, industria, educación y sistemas interactivos

Salud (healthcare)

- Radiología: apoyo a lectura (triage), reportes preliminares.
- Patología: conteo/segmentación asistida.
- Documentos clínicos: extracción estructurada y resumen.
- Riesgos: sesgos, trazabilidad, privacidad.

Educación

- Tutoría con imágenes (diagramas, pizarras).
- Corrección guiada y feedback multimodal.
- Laboratorios virtuales (pasos + verificación).

Industria

- Inspección visual: defectos, calidad, metrología.
- Mantenimiento predictivo (imagen + señales).
- Robótica/operaciones: percepción + instrucciones.
- Búsqueda en catálogos: similitud y retrieval.

Sistemas interactivos

- Asistentes con cámara (soporte técnico).
- AR/VR: instrucciones contextuales.
- Accesibilidad: describir escenas/objetos en tiempo real.