



PROBLEMAS DE **SEGMENTACIÓN**

ESTRUCTURA DE LA CLASE

INTRODUCCIÓN

Presentación del tema de la clase

Introducción a la segmentación

MECANISMOS DE SEGMENTACIÓN

Segmentación supervisada

Segmentación no supervisada

Casos de uso

REDES CONVOLUCIONALES

Autoencoders

Modelos

PÉRDIDA

Significado y usos

Funciones de pérdida para clasificación

Funciones de pérdida para regresión

Detención por pérdida

CODIFICACIÓN DE LA INFORMACIÓN

Imágenes y vídeo

Música y sonido

Texto

Olores y sabores

ACTIVIDAD PRÁCTICA

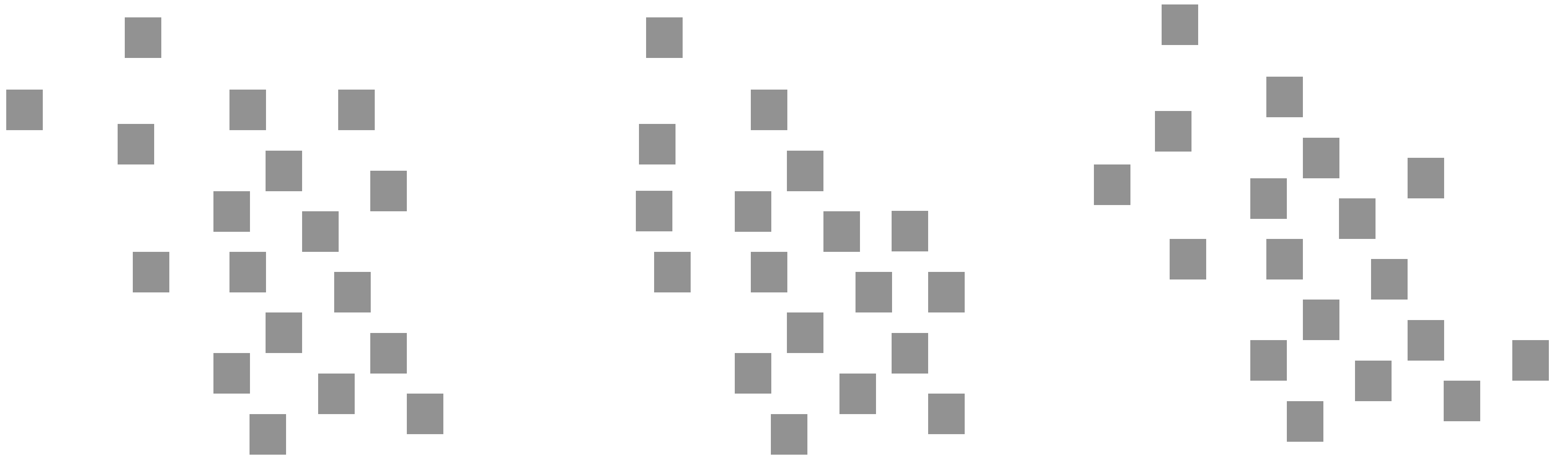
RemoveBG

Uso de modelos por API

CONCLUSIONES

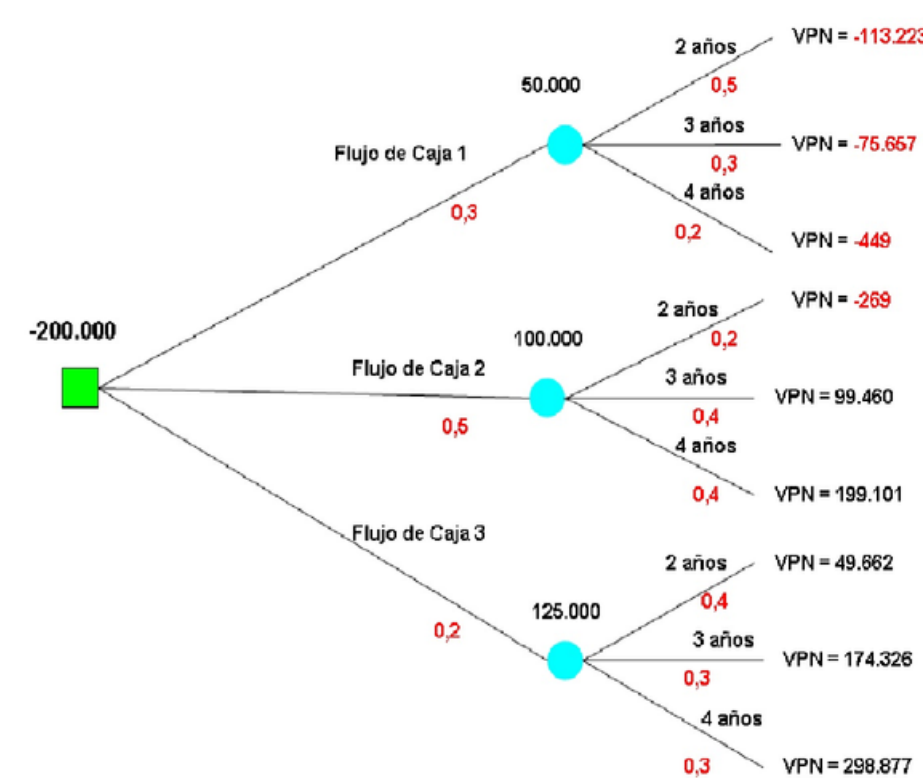
Recapitulación de los puntos clave de la clase

PROBLEMAS DE SEGMENTACIÓN

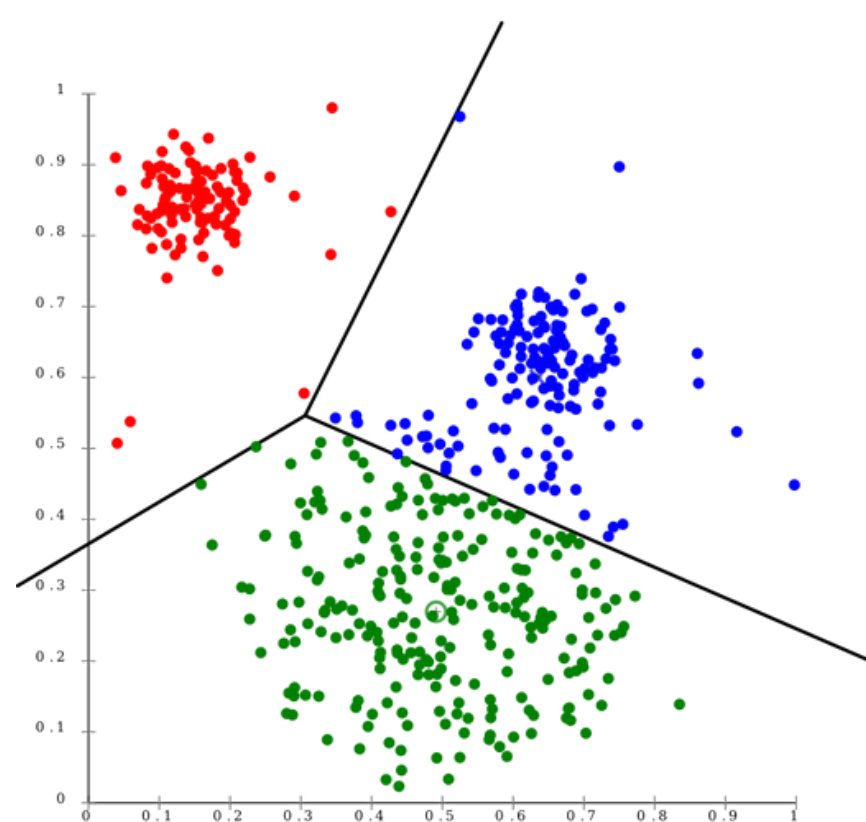


SEGMENTACIÓN SUPERVISADA

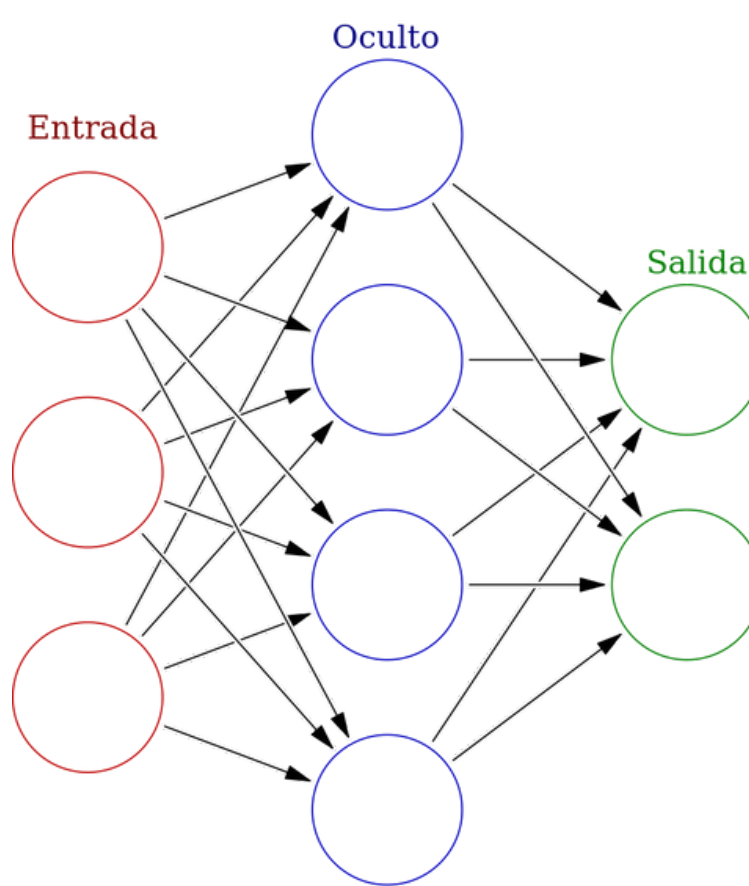
ÁRBOLES



CLUSTERS

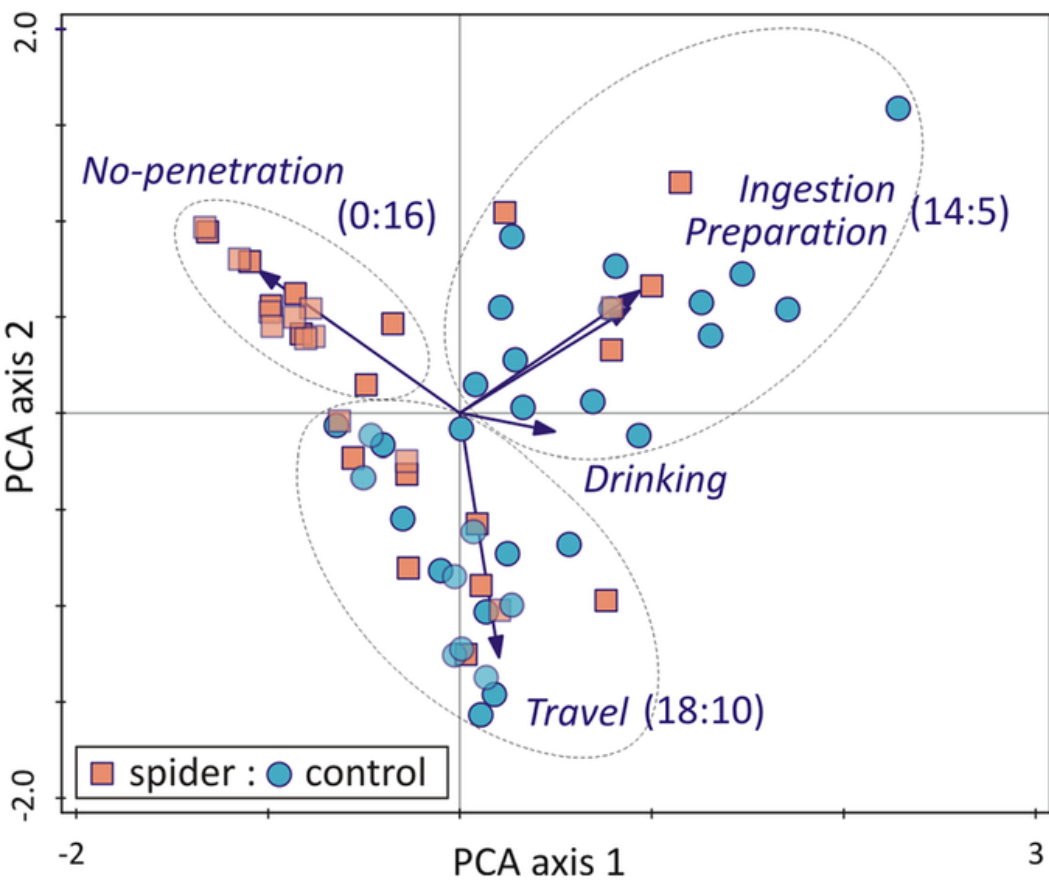


DNNs

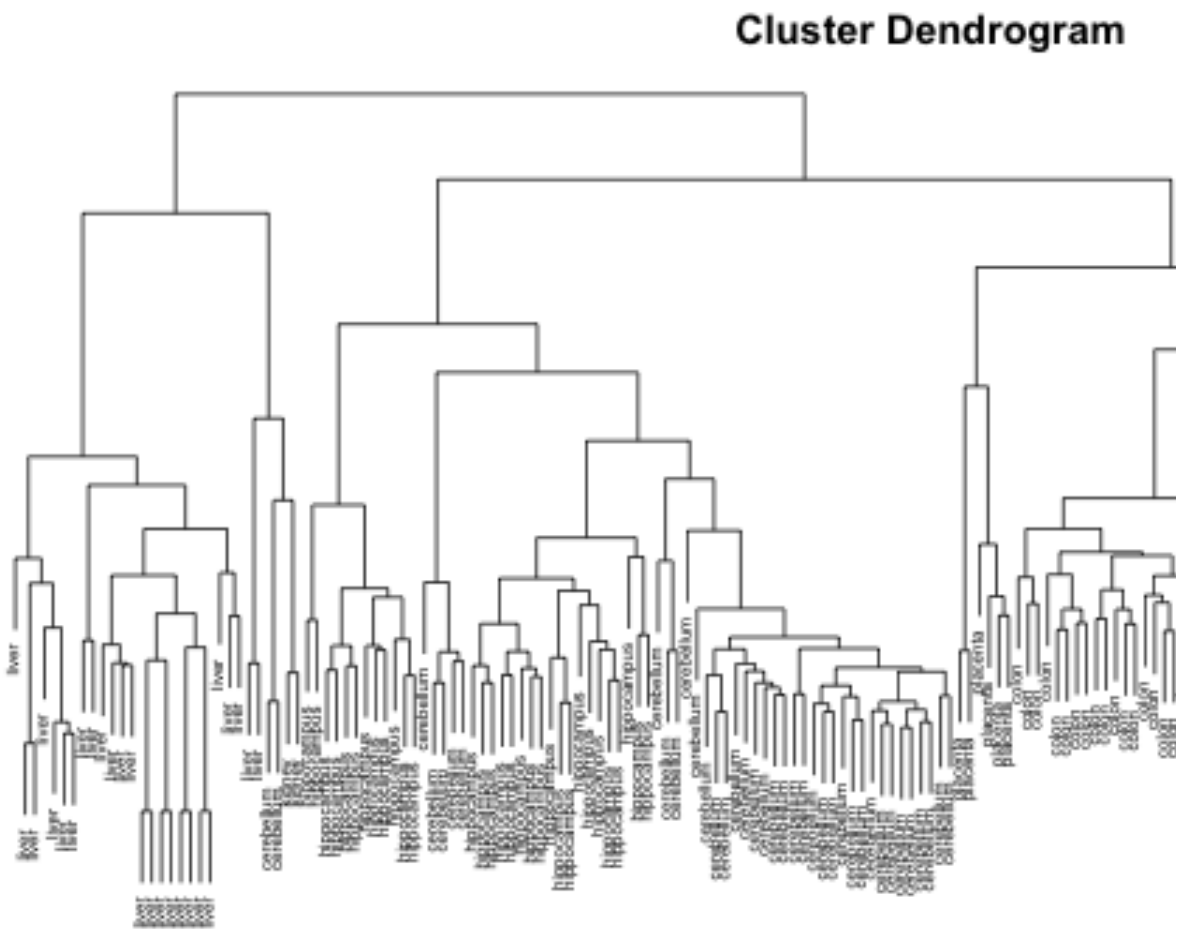


SEGMENTACIÓN NO SUPERVISADA

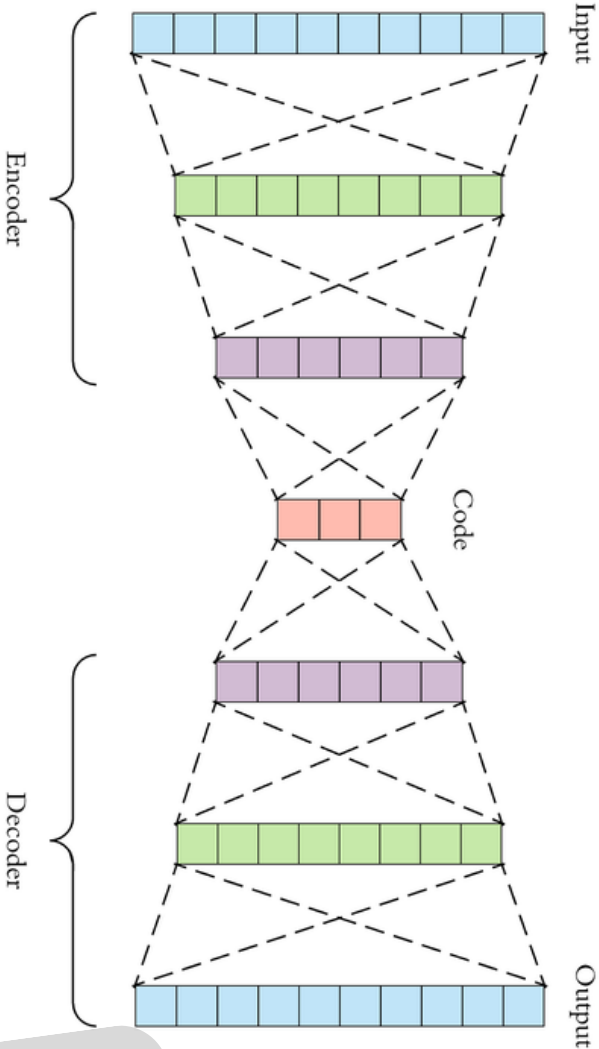
PCA



CLUSTERS



AUTOENCODERS



CASOS DE USO

AGRUPAR

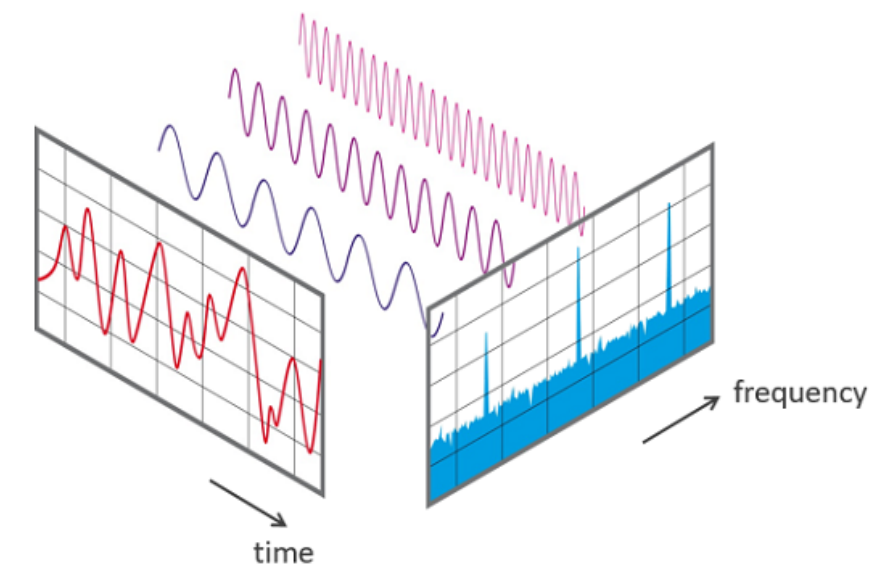
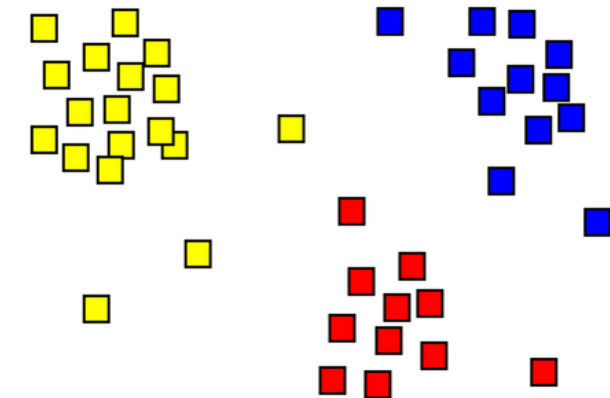
Encontrar grupos con características similares

SEPARAR

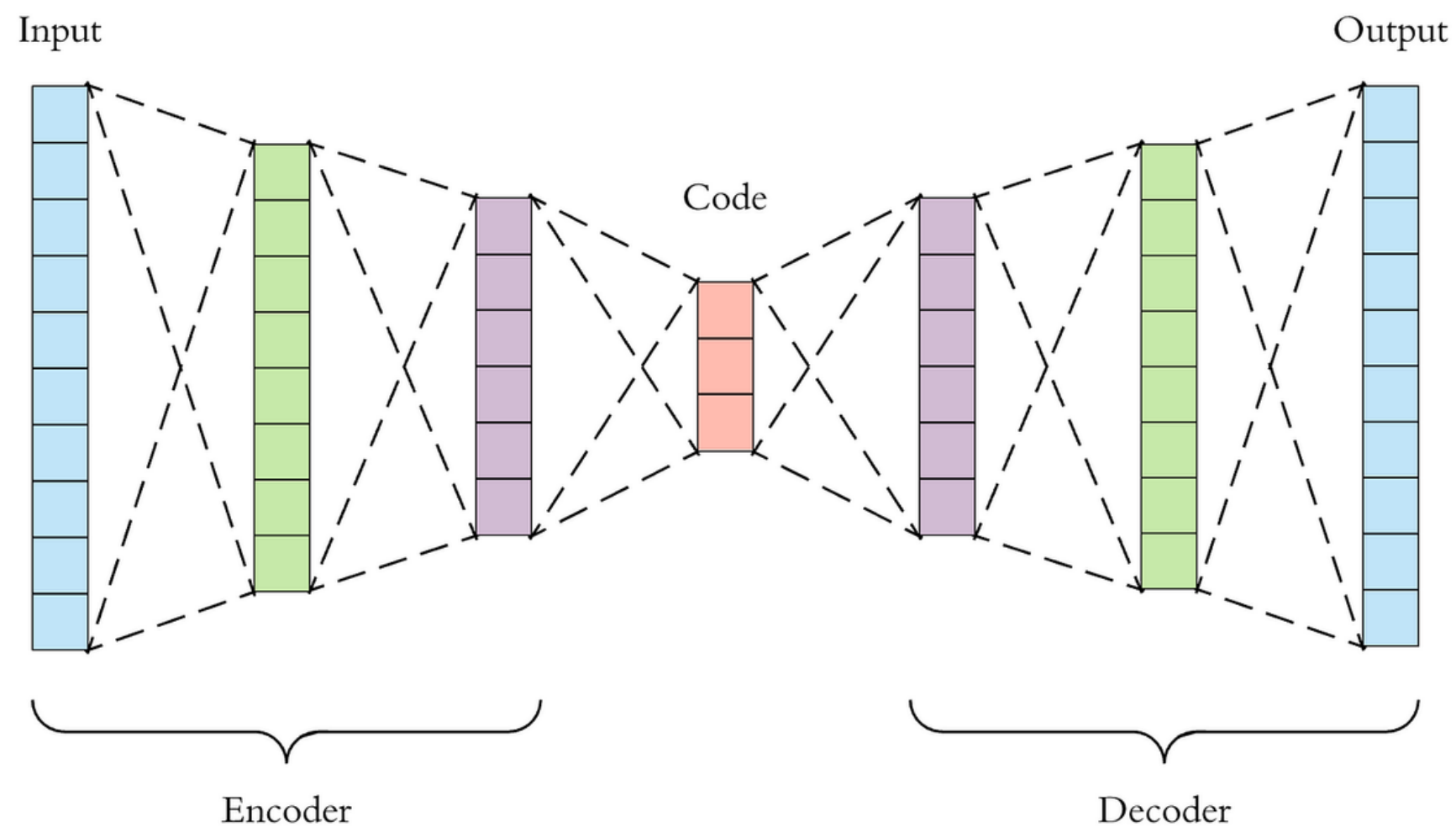
Separa elementos distintos, para trabajarlos por separado

DETECTAR IMPOSTORES

Encuentra elementos fraudulentos



AUTOENCODERS



**No buscan clasificar o predecir.
Buscan es dada una entrada
generar la misma salida**

**Al “aprender” a generar sus
entradas, el modelo “aprende”
las características**

**Se compone un codificador y
un decodificador**

MODELOS AUTOENCODERS

UNET

Creado para la segmentación de imágenes en medicina, su potente arquitectura se extendió a casos de uso como modelos de difusión

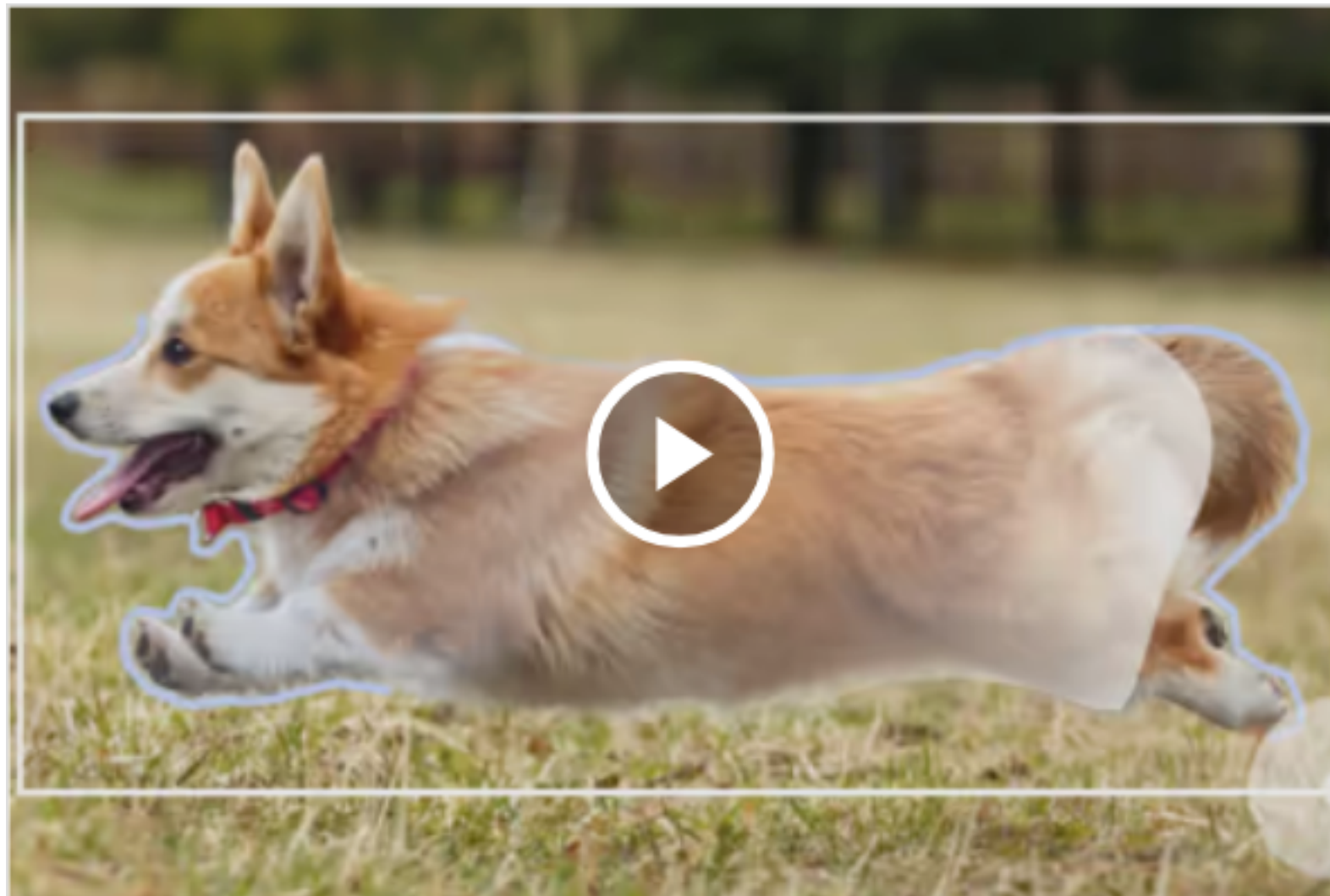
ENET

Segmentación de imágenes en tiempo real con baja latencia y alta precisión

SAM

Lanzado por Meta en abril de 2023, es el modelo más potente y usado por la comunidad. Puede segmentar prácticamente lo que sea en tiempos muy cortos

SEMÁNTICOS

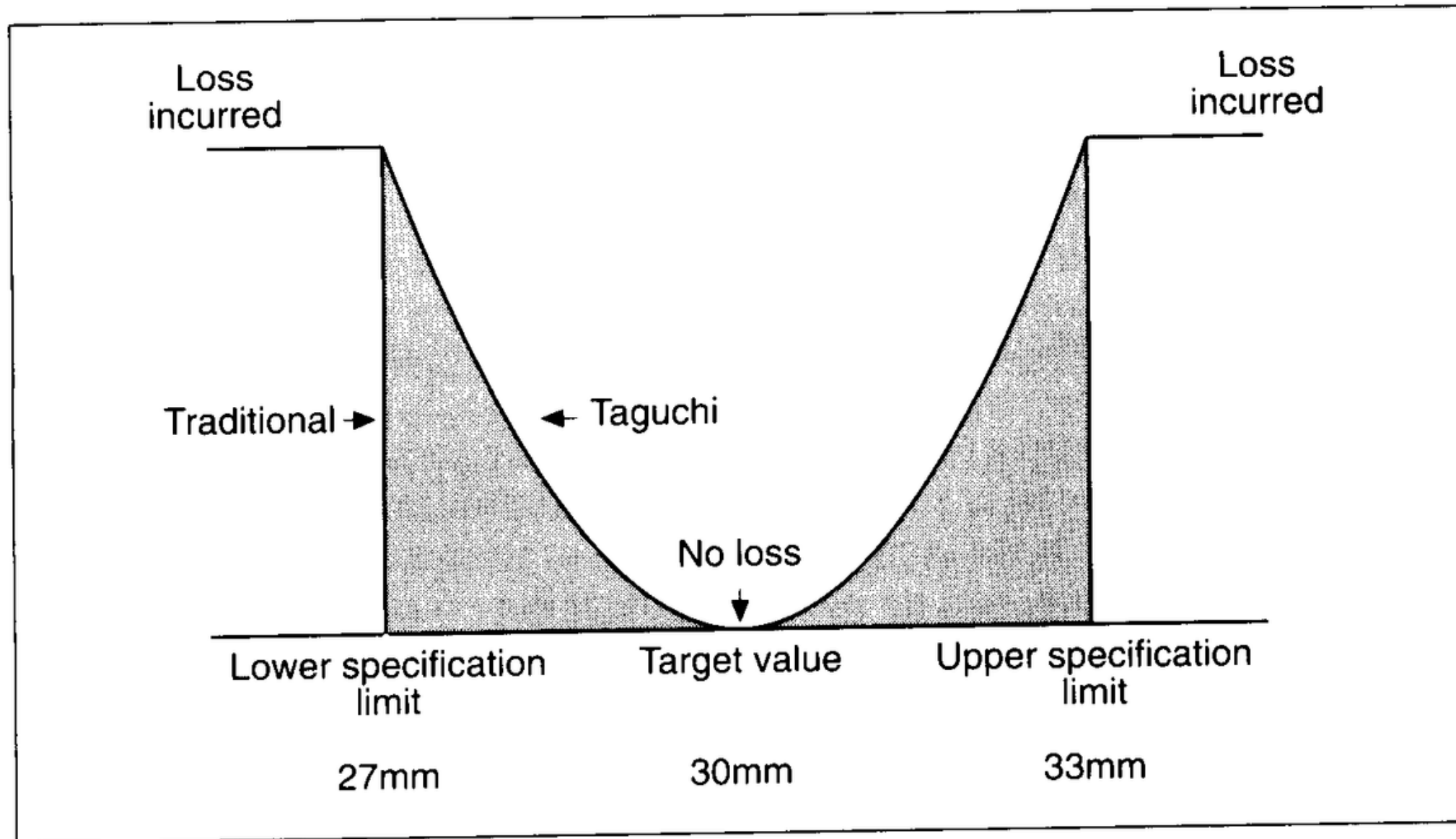


Segment Anything

Meta AI Computer Vision Research

segment-anything.com

FUNCIONES DE PÉRDIDA



Cuantifica el error entre los valores obtenidos y los deseados

Permite determinar qué tan bien ha o está entrenando un modelo

El objetivo de todo modelo es minimizar el valor de la función de pérdida

PÉRDIDA EN CLASIFICACIÓN

CLASIFICACIÓN BINARIA

La función de entropía cruzada **binary_crossentropy**, bisagra **hinge** (en SVMs)

CLASIFICACIÓN MULTICLASE Y MULTIETIQUETA

La función de entropía cruzada categórica **categorical_crossentropy**

Si no se quiere usar entropía cruzada, se puede ir, aunque no se recomienda, por el error medio absoluto **mae**, o por el cuadrático medio **rms**

PÉRDIDA EN REGRESIÓN

EN LA MAYORÍA DE CASOS

El error cuadrático medio **mse** es el más popular porque es sensible a valores atípicos

SI LOS VALORES SON MUY VARIABLES

La sensibilidad afecta a la función, entonces o se usa error absoluto medio **mae**, error cuadrático medio logarítmico **msle**

También existen la función **huber** (MSE + MAE) y de divergencia probabilística **kld**

DETENCIÓN POR PÉRDIDA



```
from keras.callbacks import EarlyStopping

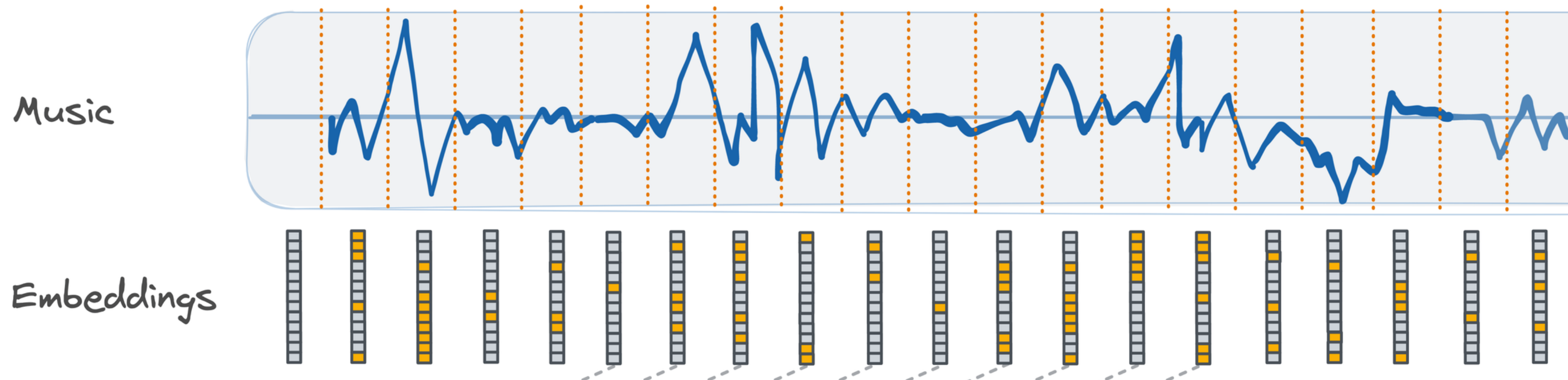
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
model.fit(x_train, y_train, epochs=10, validation_data=(x_val,
y_val), callbacks=[early_stopping])
```


CODIFICACIÓN DE IMÁGENES

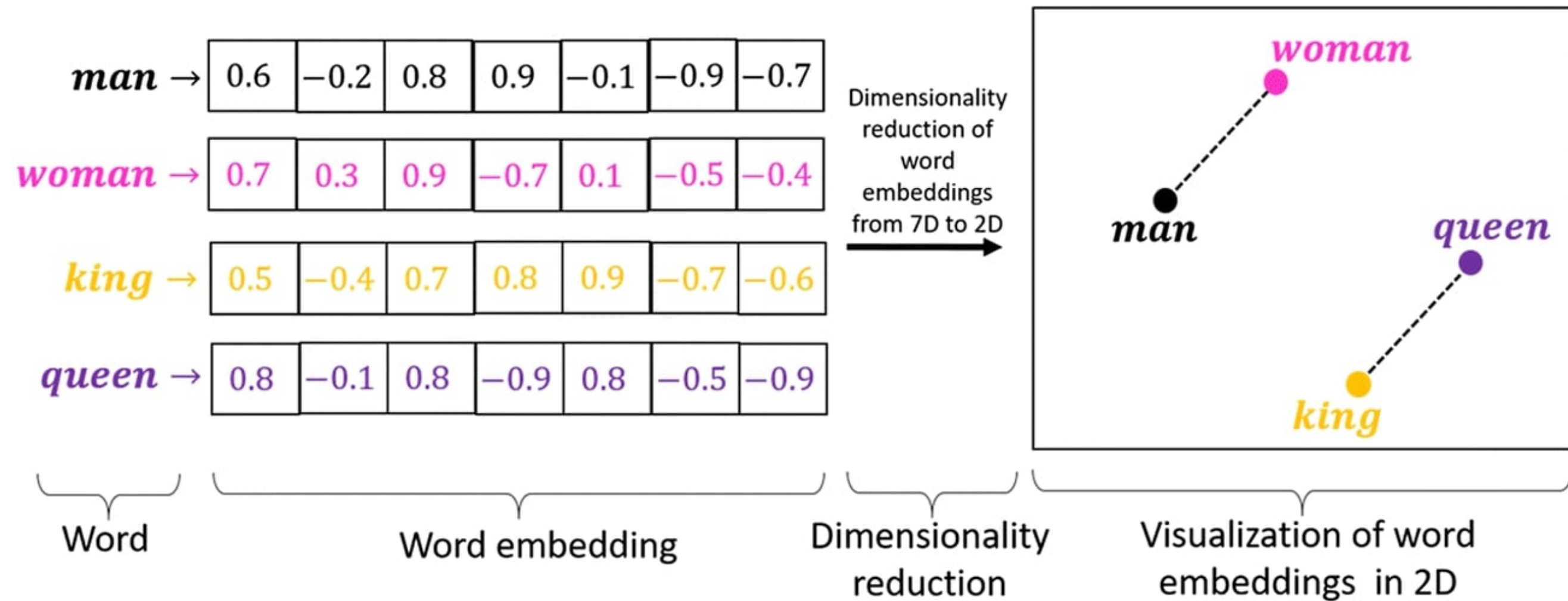


0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

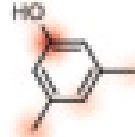





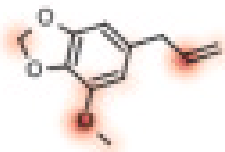
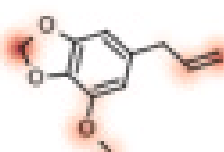
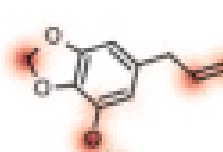






CODIFICACIÓN DE SONIDO



CODIFICACIÓN DE TEXTO



CODIFICACIÓN DE OLORES Y SABORES

CAS	Scale Feature	Vector Feature	Merge Feature
108-68-9			
1240841-77-3			
607-91-0			
106-33-2			
219691-94-8			

CONCLUSIONES

SEGMENTACIÓN

La segmentación permite separar o agrupar un conjunto de elementos con características definidas

FUNCIONES DE PÉRDIDA

El fin de todo modelo es minimizar la función de pérdida. Usamos `binary_crossentropy` y `categorical_crossentropy` en clasificación y, `mse` y `mae` en regresión

CODIFICACIÓN

Los embeddings permiten entrenar modelos que por naturaleza solo aceptan números, con prácticamente cualquier elemento de la realidad