
Robótica de Servicio

José María Cañas

josemaria.plaza@urjc.es



Grado Ingeniería Robótica Software, Curso 2022-2023

Aprendizaje Automático en Robótica de Servicio

Contenidos

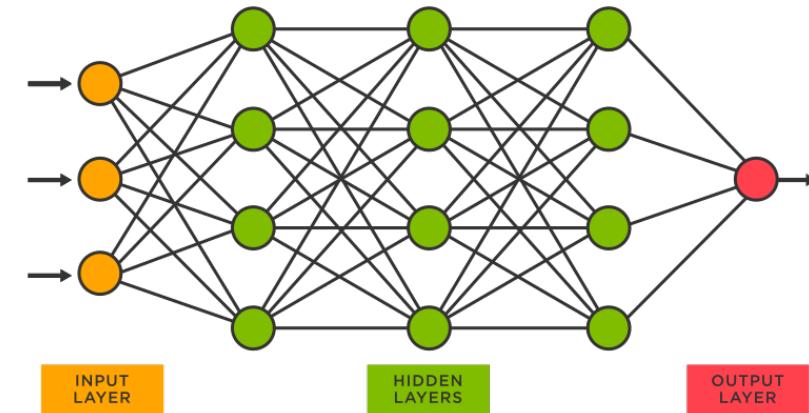
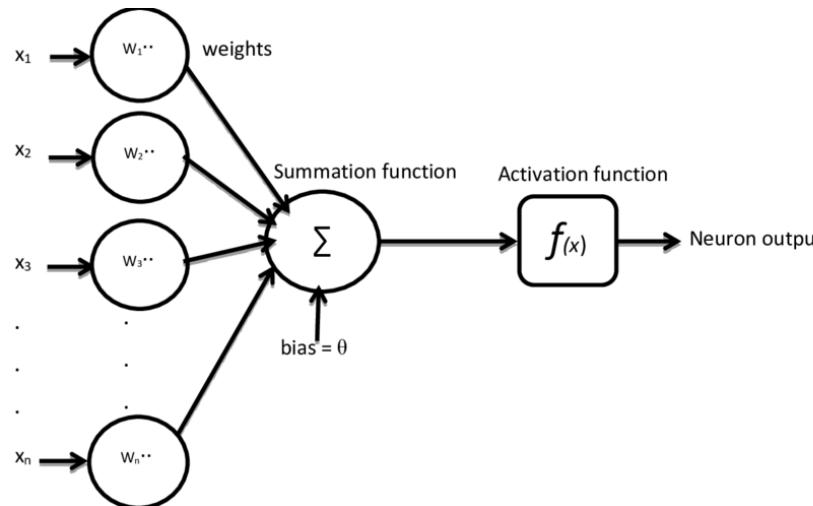
- Introducción
- Aprendizaje profundo
 - en percepción
 - en control
 - en sistemas extremo a extremo
- Aprendizaje por refuerzo

Introducción

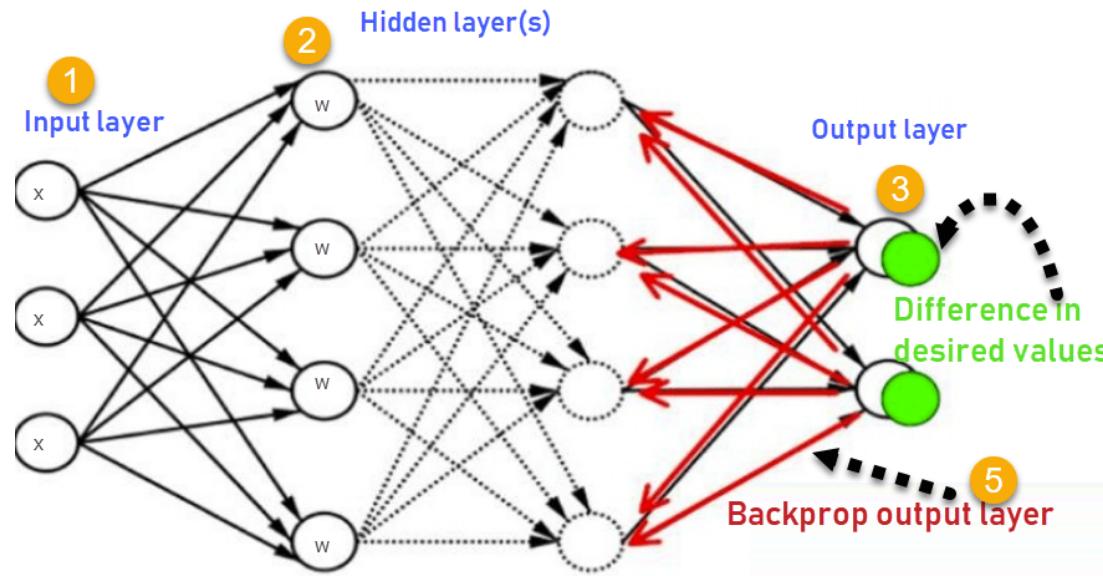
- Maneras indirectas de programar robots de servicio, no explícitas
- Aprendizaje automático
- Aprendizaje neuronal
- Aprendizaje por refuerzo
- Entrenamiento \neq inferencia
- El sistema entrenado se conecta a los sensores y o actuadores



Aprendizaje neuronal, aprendizaje profundo



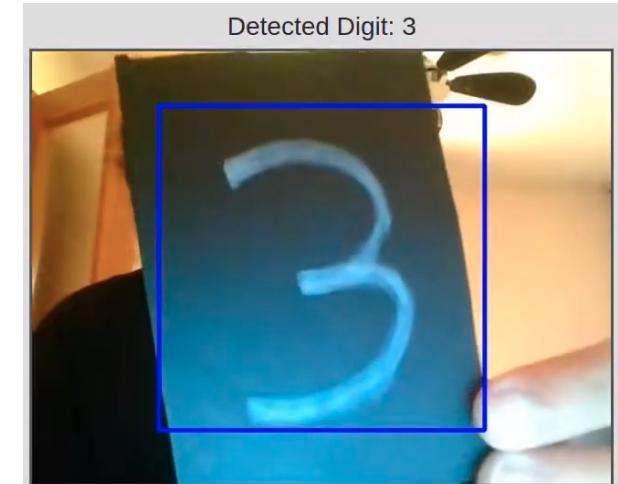
- Modelo de cómputo que se puede ajustar con datos
- 1 neurona, sumatorio ponderado, función de activación
- Red de neuronas conectadas según cierta topología (modelo) y con unos pesos
- Topologías: capas, fully connected...



- Entrenamiento = ajuste de pesos con (muchos) datos supervisados
- Algoritmo aprendizaje: *backpropagation*
- Middlewares neuronales: TensorFlow (Google), PyTorch (Meta)
- Formato standard ONNX (topología y pesos)

¿PARA QUÉ SE USAN?

- Clasificación, detección, regresión
- Aproximadores universales de funciones desde valores puntuales
- Predicciones
- ¿Y en robots?
 - en percepción, detección de objetos
 - en control, decisiones a los actuadores



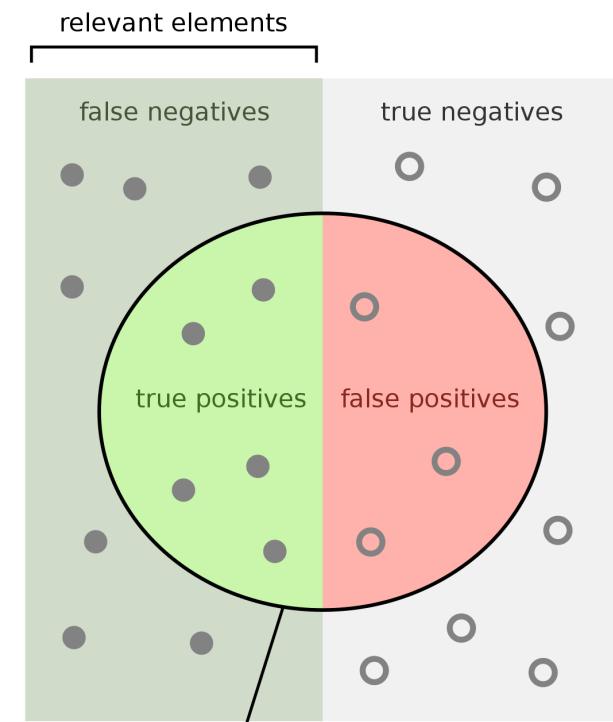
Redes neuronales para percepción de robots de servicio



- Detección de objetos en imágenes
- Mayor entendimiento de la escena del robot de servicio
- Modelos: SSD, YOLO, Faster R-CNN, Mask R-CNN...
- Redes preentrenadas, listas para usar (p.e. Darknet-ROS)



- Datasets COCO, ImageNet...
 - Imágenes etiquetadas
 - Training, validation, testing
 - Métricas de calidad:
precision, recall
- cuán bien lo hacen con el dataset de test



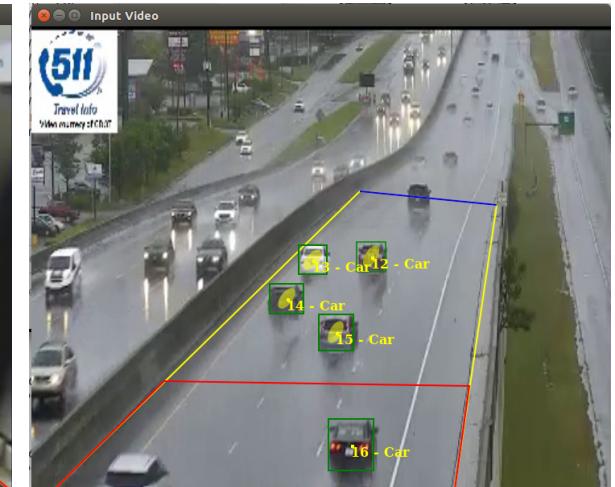
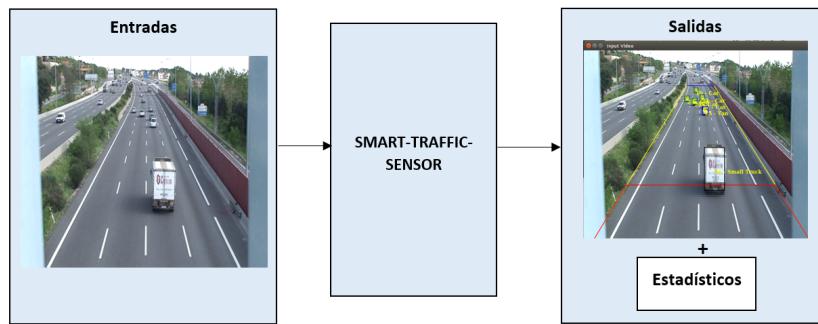
How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

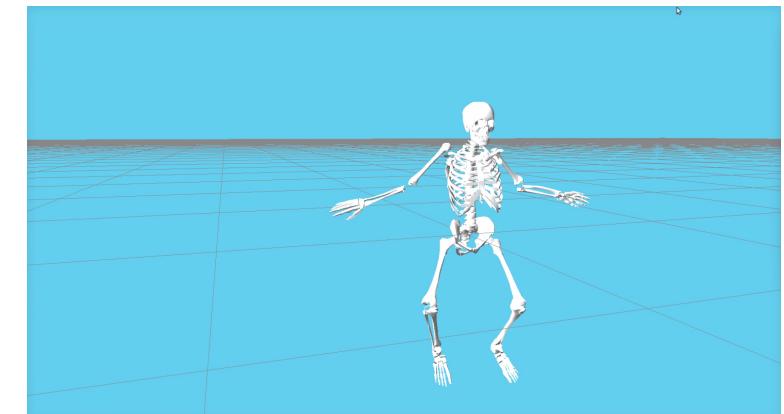
$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

MONITORIZACIÓN VISUAL DE TRÁFICO



- Detección y clasificación mejor que Visión Artificial clásica
diferencia con fondo, tracking puntos interés, clasificador SVM
- *DeepLearning*: +Robustez frente a cámaras malas y climatología
- Dataset incluye ejemplos de imágenes difíciles

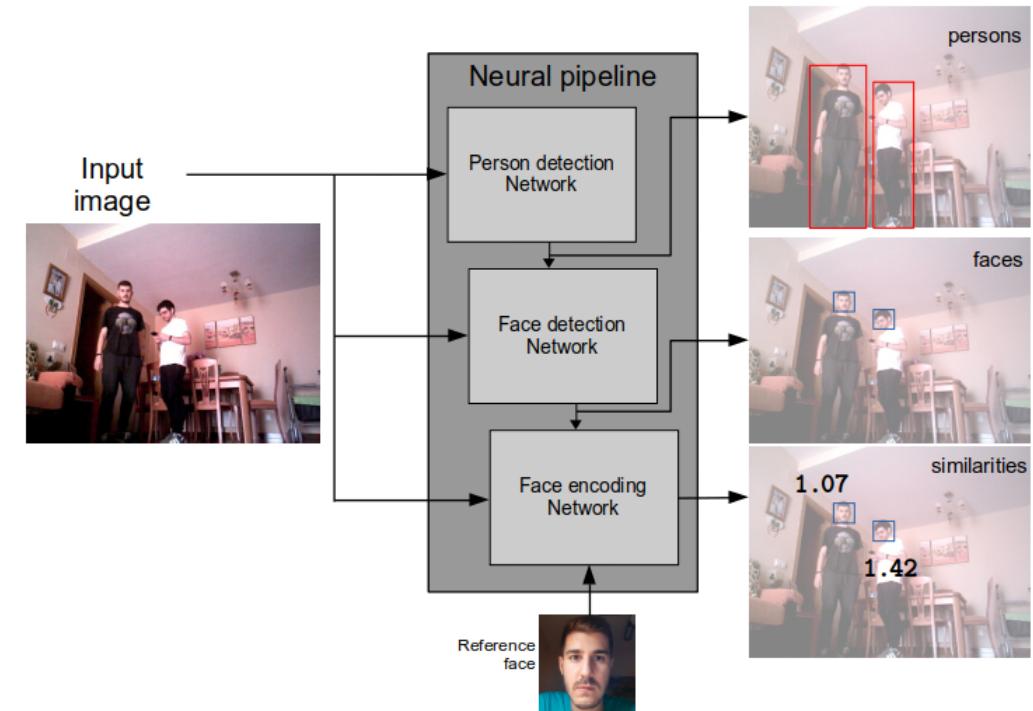
DETECCIÓN 3D DE PERSONAS, HRI



- Cámara RGBD
- *DeepLearning*: Modelo de persona 2D
(p.e. OpenPose)
- Modelo de persona 3D
- Sin marcadores IR

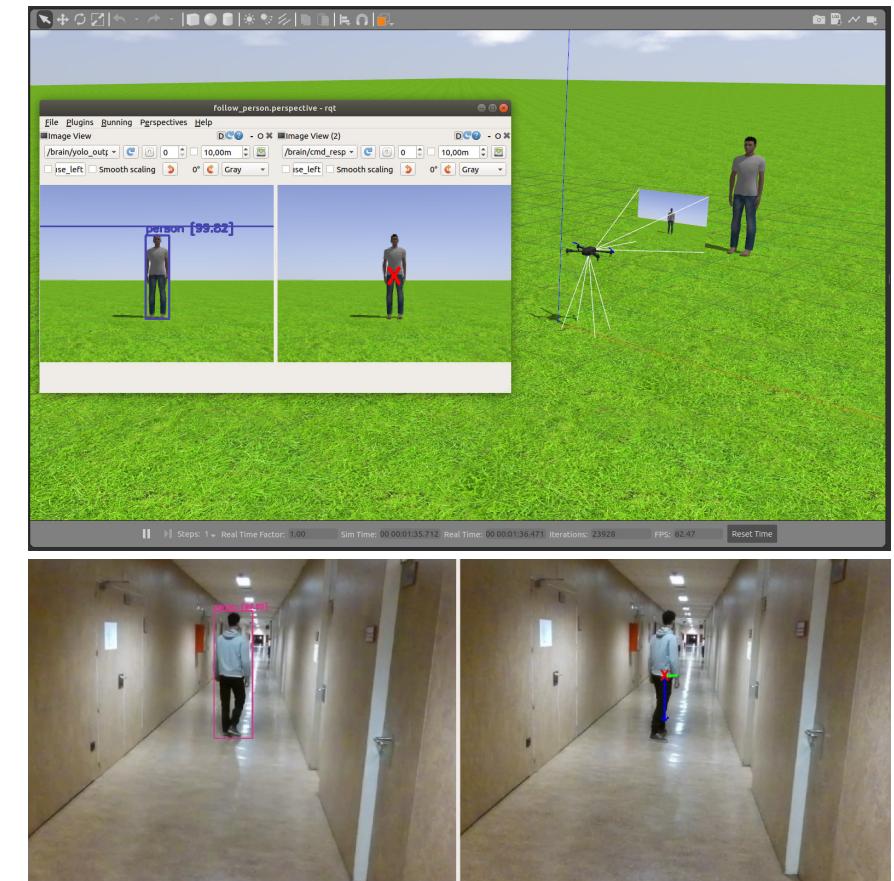
ROBOT MÓVIL SIGUE A PERSONA

- Cámara como sensor principal
- Red detecta personas
- Red detecta cara
- Red identifica individuo
- Optimizadas con TensorFlow
- Memoria de corto plazo
- Control: VFF y casos
- Seguimiento robusto



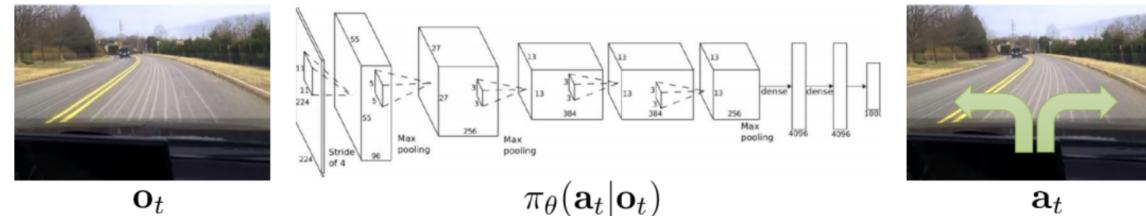
DRONE SIGUE A PERSONA

- Cámara como sensor principal
- Red neuronal detección de persona
- Control 3 PID simultáneos:
avance, giro horizontal, elevación
- Seguimiento robusto



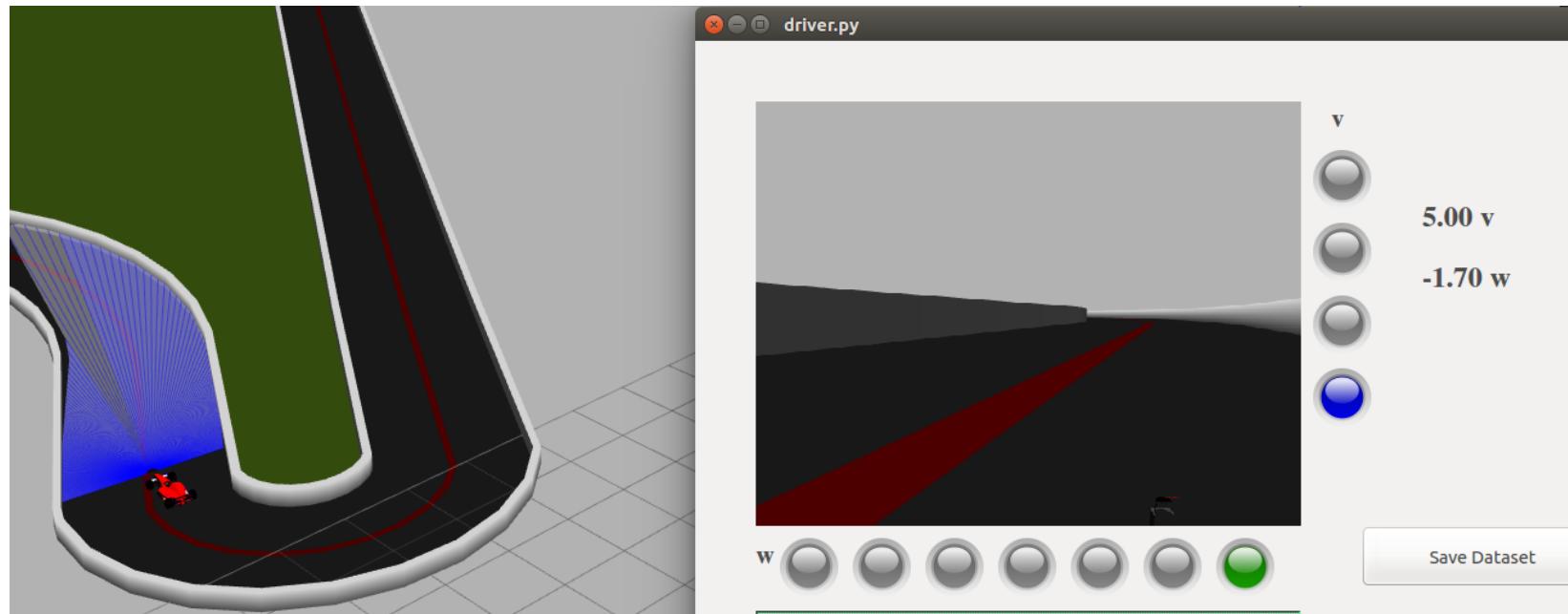


Redes neuronales extremo a extremo en robots de servicio

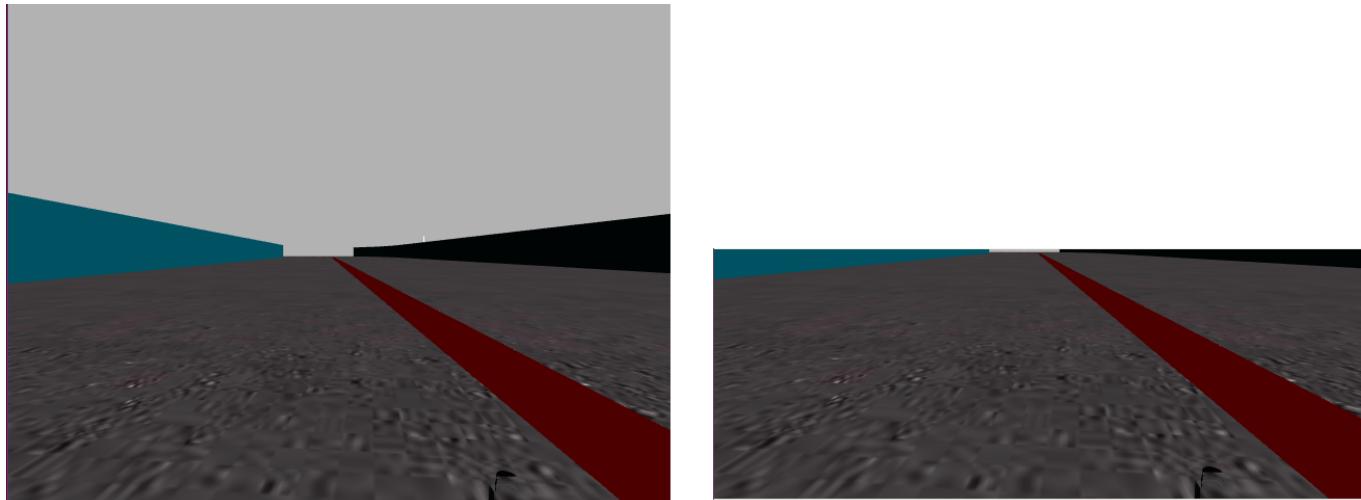


- En sistemas extremo a extremo, incluye percepción y decisiones
- Modelos y datos
- Conducción autónoma: PilotNet (nVidia)...
- Un solo fallo en el control puede ser dramático
- Frecuencia alta mejora la calidad del comportamiento generado

COCHE AUTÓNOMO SIGUE LÍNEA



- Gazebo, Formula1, línea roja
- Cámara a entrada de la red. Salida a los motores
- Dataset generado con piloto programado explícitamente o manual
- Aprende controlador reactivo (*visual!*) desde ejemplos supervisados

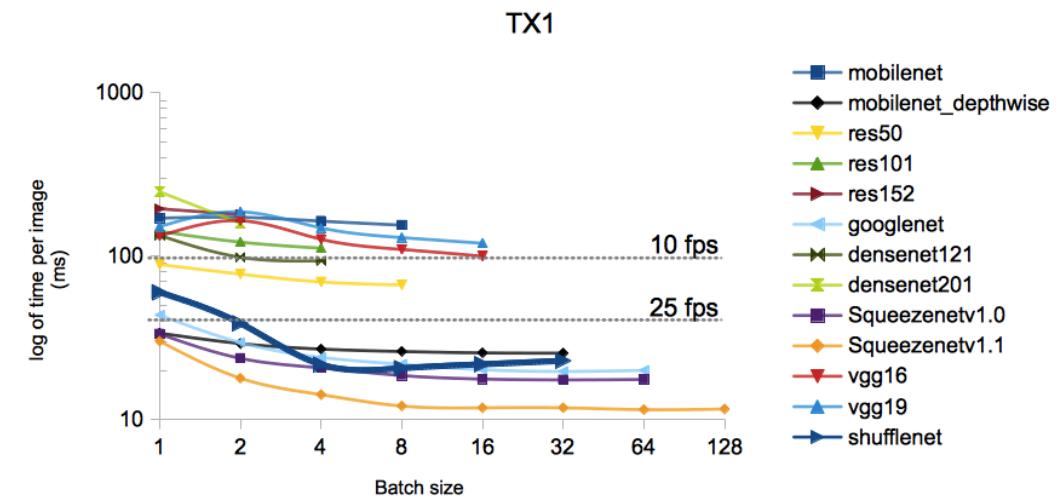
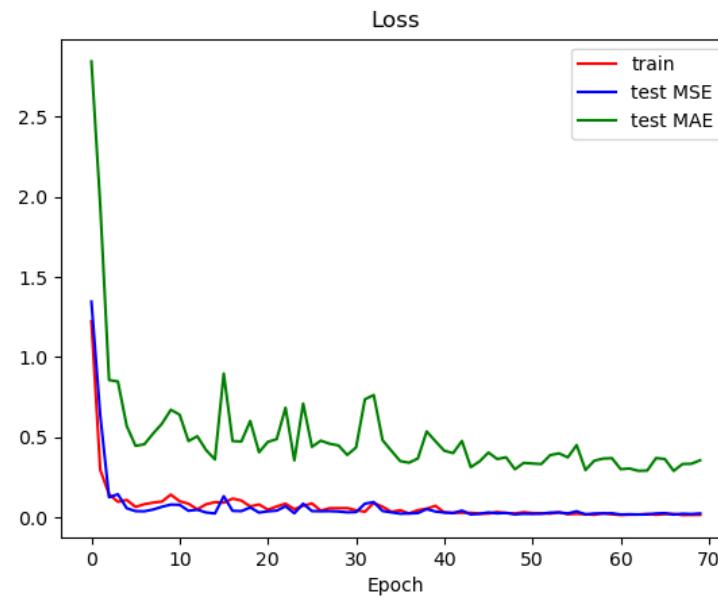


- A la primera no daba bien las curvas
- Casos difíciles aportan mucho al entrenamiento pero son poco frecuentes → aumentarlos artificialmente, dataset balanceado
- Red de clasificación: discretización de V y W
- ¿Cuántos niveles de discretización?
- ¿Imágenes recortadas? No le distraen
- Red de regresión, salidas lineales

COCHE AUTÓNOMO REAL SIGUE LÍNEA

- JetBot, Cámara y motores
- Red preentrenada se reentrena
- Distintas iluminaciones, circuitos...
- ¡Funciona!

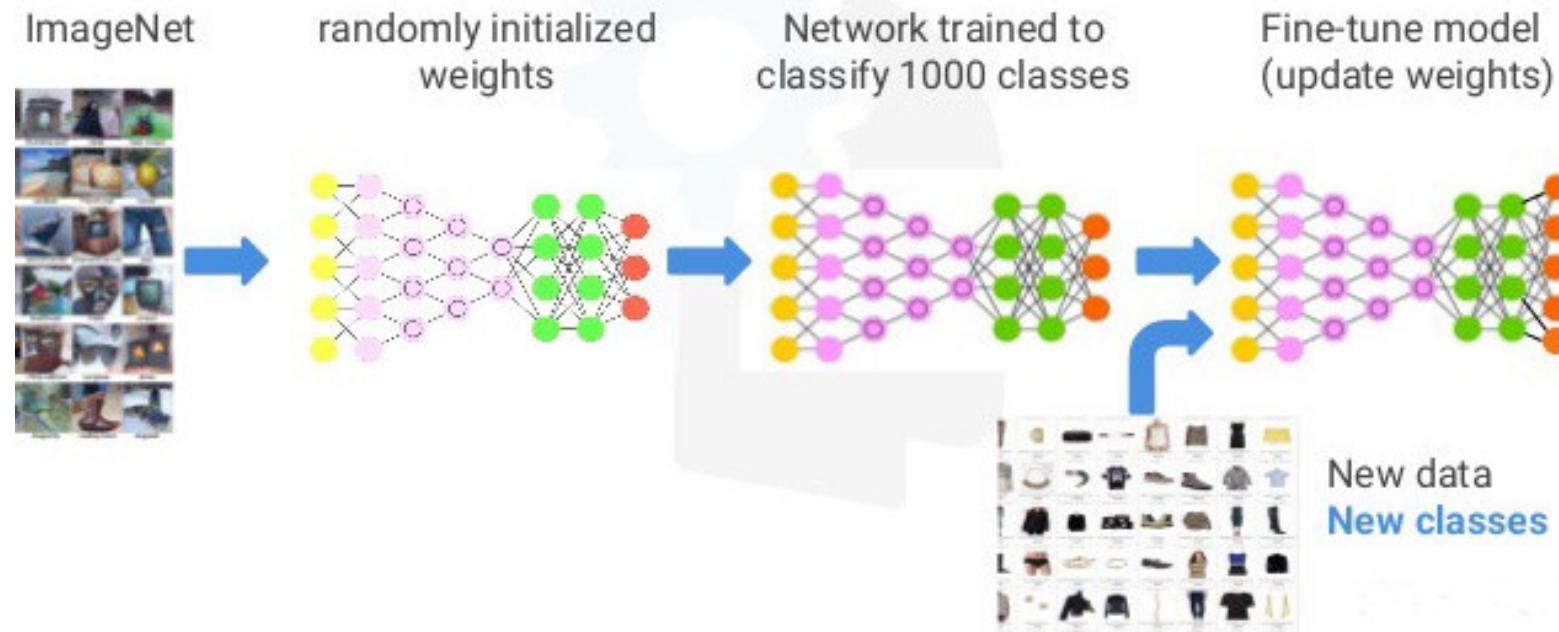




- ¿Cuándo detener el aprendizaje?
- Evolución de la función de pérdida
- Tiempo de inferencia es crítico
- Hay unas redes más rápidas que otras



Transfer learning



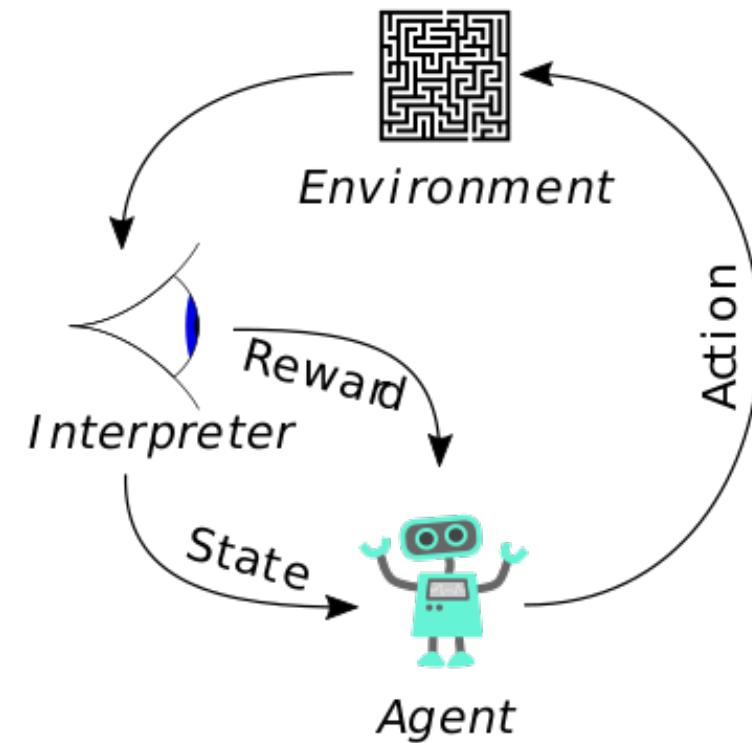
- (Re)entrenar redes preentrenadas adaptándolas a problemas concretos
- El reentrenamiento parte de unos pesos razonables
- Reutiliza cálculo de descriptores interesantes (*features*)

Balance general

- Aportan robustez, imprescindible en mundo real
- Generalizan razonablemente bien (con límites, no hacen magia)
- En general lentas, coste computacional
 - Se pueden optimizar: CPU, GPU, chips de AI
- Poca explicabilidad
- Modelos: redes convolucionales (CNN), redes recurrentes, LSTM...
- Datasets supervisados, etiquetado
 - Generación con simuladores o etiquetando a mano (LabelMe)
 - Cuanto más datos y más variados, mejor. Volúmenes grandes
 - *Data Augmentation*

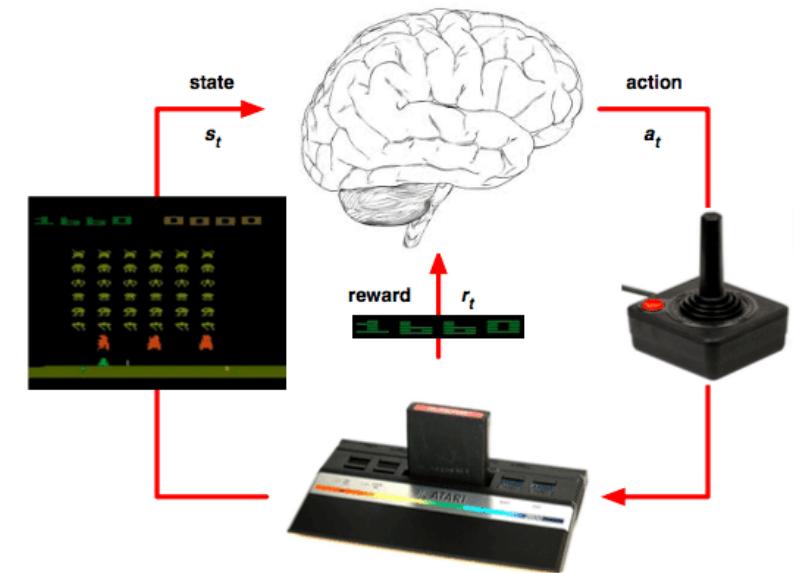
Aprendizaje por refuerzo

- Ensayo y error, muchos intentos se apoya en simulación
- Agente y ambiente
- Acción, estados, recompensas
- Algoritmos de aprendizaje: Q-Learning, DQN...
- Asignación de crédito
- ¿En robots de servicio?: control
- OpenAI: Multi-Agent Hide and Seek

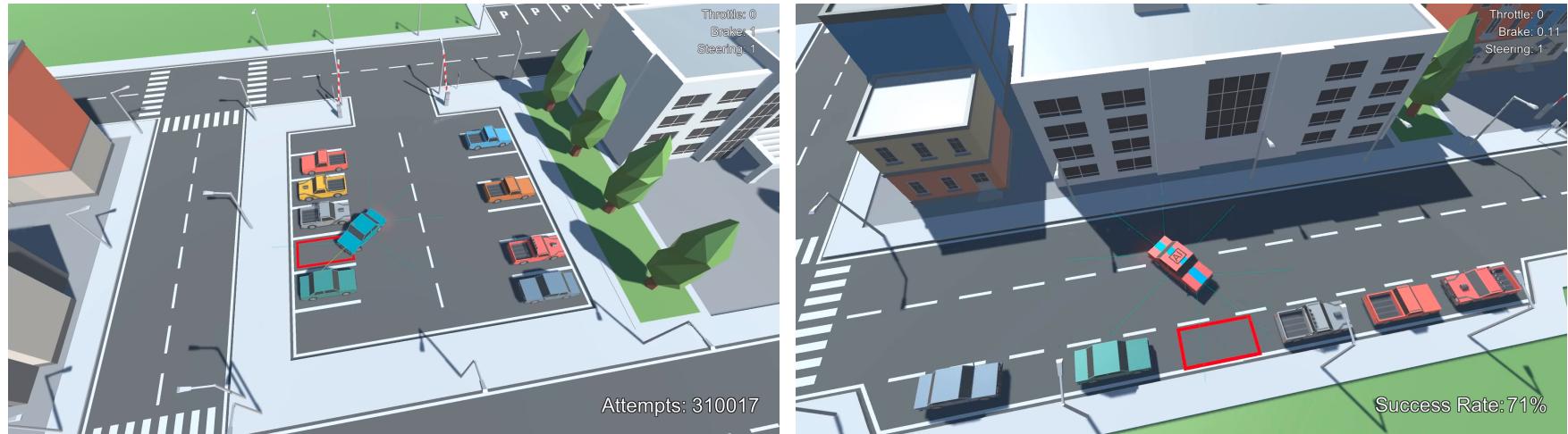


VIDEOJUEGOS DE ATARI

- Empresa DeepMind (Google 2014)
- Entrada puramente visual
- Puntos tal y como son en el juego Breakout
- Jugador automático
- Algoritmo: Deep Q Learning
- Agent57 (2018)



AUTOPARKING



- Control para aparcamiento automático
- Distintas posiciones de inicio y finales
- Recompensa: llegar y ajustar, no chocar
- En batería
- En línea

SEGUIMIENTO DE UNA LÍNEA BASADO EN VISIÓN

- Coche Formula1 con cámara
- Puntos de inicio aleatorios
- Recompensa por tener centrada la línea roja en imagen
- Funciona

