

USO DE REDES NEURONALES EN LA ROBÓTICA MÓVIL

INTRODUCCIÓN

Este trabajo pretende explorar soluciones implementadas con redes neuronales para diferentes aspectos de la robótica móvil, incluyendo planificación de trayectorias, exploración autónoma y navegación basada en sensores.

La exploración de espacios desconocidos es un requerimiento fundamental para un robot móvil y para tareas como limpieza, operaciones de rescate, navegación, exploración de espacio, etc. En la robótica móvil clásica, la exploración normalmente se basa en un modelo probabilístico, mapas de coste e información geométrica. No obstante, la mayoría de estos enfoques no utilizan ningún proceso cognitivo y trabajan a nivel de inteligencia bajo. Para mejorar este aspecto se emplea deep learning para simular el comportamiento humano frente a una tarea de exploración de espacio desconocido.

EXPLORACIÓN AUTÓNOMA

Vista desde el punto de vista de las redes neuronales la exploración autónoma se puede dividir en dos partes: visión por computador y toma de decisiones. Para ambas partes se emplea una red diferente debido a que el resultado deseado no es el mismo.

- **Visión por computador**

Se busca que el robot reconozca el espacio en el que se encuentra para poder basar su decisión futura en este conocimiento. Se suelen usar *redes neuronales convolucionales (CNN)* para este paso ya que son capaces de extraer características claves de los espacios de forma similar a un humano. En la figura 1 se puede observar el proceso típico de extracción de características por una CNN. Las convoluciones usadas en estas redes se pueden considerar filtros especiales para las imágenes recibidas por la cámara.

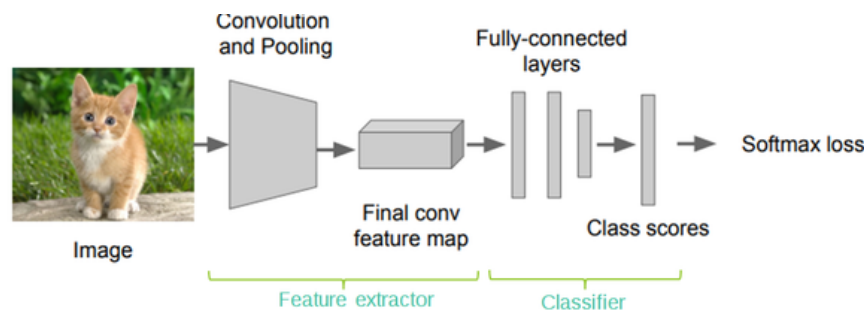


Figura 1: el esquema básico de una red neuronal convolucional

La aplicación de capas de convoluciones es importante para el reconocimiento ya que pueden determinar características importantes de los espacios, como las esquinas, paredes, puertas, etc. Si la red detecta 2 esquinas y una pared puede concluir que es una "calle ciega" para el robot y pasar esta información a la red neuronal responsable de toma de decisiones. En la figura 2 se puede observar que la CNN dada usa varias capas convolucionales para extraer las características claves vistas en la parte derecha de la figura.

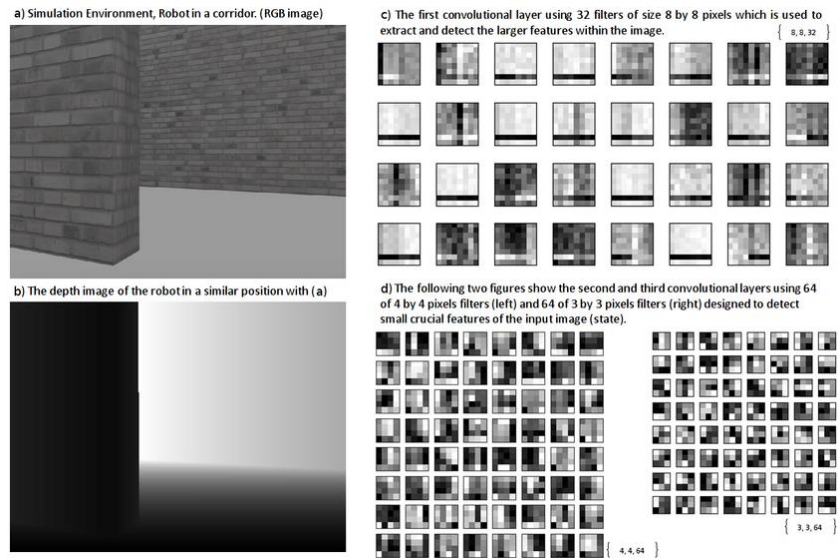


Figura 2: extracción de características de un espacio por una CNN

Después de sacar dichas características se le aplica a cada capa una función de activación. Las funciones de activación sirven para ajustar una recta al conjunto de datos para separarlo en dos, tres, etc. Las funciones de activación pueden ser distintas. En las tareas de clasificación entre 2 clases se suelen usar funciones de activación lineales, ya que es fácil separar un conjunto de datos en 2 con una función lineal, pero en las tareas de reconocimiento de espacios por las redes neuronales no se puede usar una función de activación lineal, ya que la clasificación es mucho más compleja. En las figuras 3 y 4 se pueden observar algunas funciones aplicadas para separar conjuntos de datos.

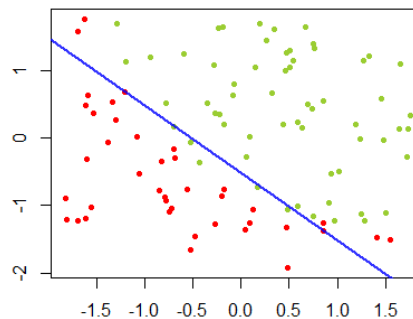


Figura 3: separación del conjunto de datos en 2 con una función lineal

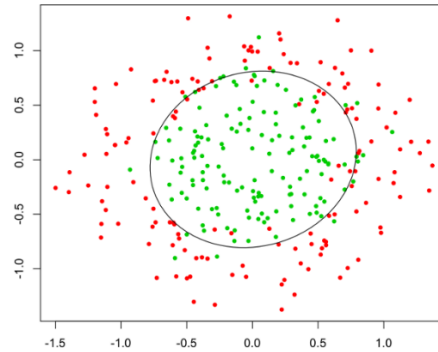


Figura 4: separación del conjunto más compleja con una función no lineal

Con las CNN se suele usar una función de activación llamada ReLu que funciona muy bien para este tipo de tareas ya que tiene en cuenta comportamientos no lineales de los conjuntos de datos. La función se muestra en la figura 5.

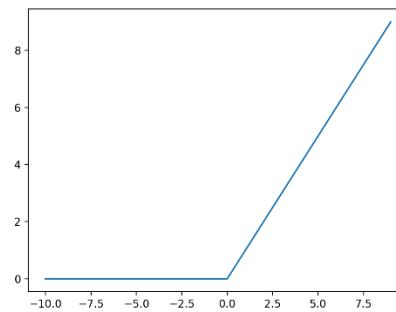


Figura 5: función de activación ReLu

Otro paso importante en las CNN es pooling. Pooling sirve para reducir el tamaño de una convolución obtenida, mejorar la robustez, eliminar ruido y quedarse con las características más importantes de la imagen.

En la figura 6 se puede observar una CNN desarrollada para un robot móvil para analizar el entorno en que se encuentra.

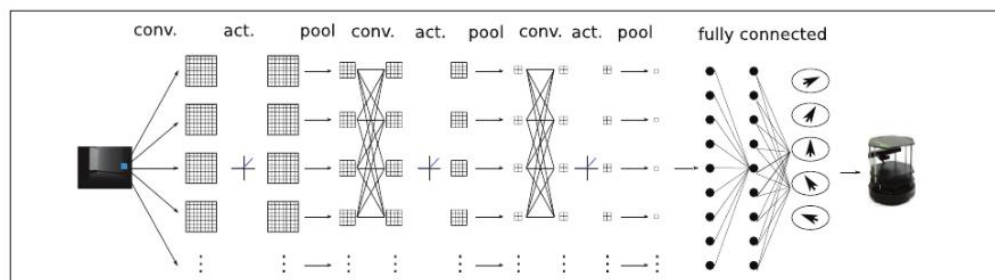


Figura 6: un modelo CNN para la exploración con 3 capas convolucionales y 2 capas conectadas

- **Toma de decisiones**

La toma de decisiones implica un nivel de inteligencia más alto. Esta parte se puede afrontar de dos maneras: emplear un clasificador discreto o clasificador basado en confianza. Para el clasificador discreto, se tiene solo una decisión y puede ser: giro derecho, medio-giro derecho, ir recto, medio-giro izquierdo, giro izquierdo, etc. Para el clasificador basado en confianza determina la confianza (probabilidad) de cada giro de la siguiente manera: si la confianza para hacer el giro derecho es de 0.3 y la confianza para seguir recto es 0.29 el robot no elige una u otra, sino calcula la velocidad angular a base de estas confianzas, resultando en una decisión más “personalizada”.

De esta manera se consigue una red similar al comportamiento humano apta para exploraciones de territorios nuevos. Es interesante que en este modelo robot no usa ningunos sensores típicos usados en robótica (laser, sonar, etc.) excepto de la cámara.

APRENDIZAJE POR REFUERZO

Una de las técnicas más importantes para la robótica móvil es el aprendizaje por refuerzo (RL). El aprendizaje por refuerzo consiste en la toma de decisiones basada en recompensa. El robot busca a aumentar la recompensa generada al seguir una trayectoria, realizar una tarea específica, etc. La elección de la recompensa correcta es muy importante para la robótica móvil y en muchas ocasiones puede dar lugar a unas situaciones inesperadas. Si se quiere que el robot alcance un determinado punto de la habitación, sería incorrecto asignarle una recompensa positiva al moverse cada metro. El robot intentará acumular la máxima recompensa y nunca llegará al objetivo, ya que simplemente moviéndose por la habitación en la dirección cualquiera le garantiza una recompensa prácticamente infinita.

Sin embargo la exploración basada en RL también presenta sus dificultades:

- **The Hard Exploration Problem:** exploración de entornos con poca recompensa o con recompensa. Es difícil porque la exploración aleatoria en tales escenarios raramente puede dar lugar a que el robot descubra estados exitosos u obtenga retroalimentación significativa.
- **The Noisy TV Problem:** el problema "Noisy-TV" comenzó como un experimento mental. Se da un agente (robot) que obtiene recompensa al buscar una experiencia novedosa. Un televisor con salidas de ruido aleatorias incontrolables e impredecibles sería capaz de atraer la atención del agente para siempre. El agente obtiene nuevas recompensas de la televisión constantemente, pero no logra ningún progreso significativo y se convierte en un “teleadicto”.



Agent in a maze with a noisy TV



Agent in a maze without a noisy TV

Figura 7: representación del problema Noisy TV ([enlace a gif](#))

A continuación se presentan algunos tipos de exploración basada en RL:

- **Recompensas intrínsecas:** para incentivar la exploración, especialmente en caso de Hard Exploration Problem, se propone aumentar la recompensa del entorno una con una señal adicional. La recompensa entonces se compone de dos términos: la recompensa que proviene de la tarea deseada y la recompensa extra que se añade cuando el agente explora.
- **Exploración basada en predicción:** las recompensas intrínsecas de exploración son dadas por la mejora del conocimiento del agente sobre el entorno. La familiaridad del agente con el entorno se puede estimar a través de un modelo de predicción. La idea de usar un modelo de predicción para medir la curiosidad se propuso hace mucho tiempo.
- **Exploración basada en memoria:** se usa memoria externa para resolver las desventajas de los previos modelos.
- **Exploración directa:** fase 1: explora hasta que se resuelve el problema, no se involucran las redes neuronales; fase 2: aumento de la robustez, el agente empieza al lado del ultimo estado y va para atrás.
- **Q-learning:** el agente aprende una serie de normas que le indican que hacer bajo ciertas circunstancias.

RL es una tecnología innovadora que tiene el potencial de transformar nuestro mundo. Sin embargo, no es necesario utilizarla en todos los casos. RL parece ser la forma más cómoda de hacer que un robot sea creativo y busque nuevas formas innovadoras para realizar sus tareas.

NAVEGACIÓN BASADA EN SENSORES

La información sensorial también se puede usar en la implementación de redes neuronales. Para ello se pueden usar *redes profundas de aprendizaje por refuerzo (DQN)*. Esas redes ayudan a los robots a encontrar trayectorias optimas desde una ubicación arbitraria sin chocar con obstáculos. En este caso los datos sensoriales se usan como entradas u observaciones y las instrucciones de giro (recto, derecha, izquierda, etc.) son salidas proporcionadas por la red.

El modelo aprende las distancias entre el robot y los obstáculos u objetivos. Si el robot choca con un obstáculo, el modelo aprende sobre ese obstáculo recibiendo una recompensa negativa basada en la distancia entre cada sensor en el robot y el obstáculo. De la misma manera, cuando el robot llega a un objetivo, el modelo aprende acerca de ese objetivo basado en la distancia a la meta y los datos posicionales de la meta mediante una recompensa positiva. Por lo tanto, no se requiere un nuevo entrenamiento si existen cambios en las posiciones de obstáculos u objetivos en un entorno dado.

Si el robot choca con obstáculos o se mueve sin colisión, el modelo aprende el entorno, incluso si no encuentra el objetivo, ya que recibe recompensas negativas basadas en colisiones con obstáculos, la proximidad del objetivo, y la distancia entre él y los obstáculos. Por lo tanto, a medida que el aprendizaje progresa, el modelo aprende el entorno basado en las recompensas. Si el modelo recibe recompensas negativas, significa que el robot y un obstáculo se están acercando entre sí, pero si el modelo recibe recompensas positivas, significa que el robot y el objetivo se están acercando.

OTROS ALGORITMOS PARA PLANIFICACIÓN DE TRAYECTORIAS

En esta sección se compararán y resumirán otros algoritmos de planificación y algunos algoritmos ya mencionados.

- **DQN:** la red neuronal aproxima la función de valor optima. Se consigue ajustando los pesos de la red. Después de terminar el entrenamiento la función de valor ya no cambiará.
- **PDQN (Potential DQN):** es una mejora de DQN. Su objetivo es aumentar la velocidad del algoritmo. Para ello se añade un campo gravitación al modelo que construye la relación entre la recompensa y la gravedad.
- **A3C:** introduce un mecanismo de evaluación y utiliza una red neuronal para predecir la acción seleccionada.
- **DDPG (Deep Deterministic Policy Gradient):** un algoritmo mejora de DQN con AC3 añadido, es una fusión de redes neuronales y aprendizaje por refuerzo.

CONCLUSIONES

Deep learning experimentó un gran avance en los últimos cinco años. Muchos problemas de robótica clásica, como SLAM y el reconocimiento visual de entornos comienzan a explorar el posible avance motivado por el Deep learning. La tarea de percepción es muy similar a la tarea de visión por computador pura y se beneficia mucho de Deep learning. Existen varias direcciones futuras en la percepción del robot para explorar: estructuras multimodales que toman información de varios sensores, convoluciones 3D para detección en coches autónomos y otros modelos como Deep Inverse Reinforcement Learning. La combinación de las políticas de control tradicionales y Deep learning puede aportar mucha conveniencia y eficiencia a las tareas robóticas.

REFERENCIAS

- [1] <https://www.quora.com/How-can-I-use-CNN-for-feature-extraction-of-images>
- [2] https://www.researchgate.net/figure/Visualization-of-our-DNN-convolutional-neural-network_fig9_328470515
- [3] https://www.researchgate.net/publication/318678743_Autonomous_exploration_of_mobile_robots_through_deep_neural_networks
- [4] <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- [5] <https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/>
- [6] <https://ieeexplore.ieee.org/document/8367110>
- [7] <https://lilianweng.github.io/lil-log/2020/06/07/exploration-strategies-in-deep-reinforcement-learning.html>
- [8] <https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/>
- [9] <https://www.frontiersin.org/articles/10.3389/fnbot.2020.00063/full>
- [10] https://www.researchgate.net/publication/311805526_Deep-learning_in_Mobile_Robotics_-from_Perception_to_Control_Systems_A_Survey_on_Why_and_Why_not