



Universidad
Tecmilenio®





Proyectos de Aprendizaje Profundo

Aplicaciones del aprendizaje profundo

Solución de tareas basadas en aprendizaje profundo



Los algoritmos de aprendizaje supervisado incluyen algoritmos que utilizan datos (ejemplos) de entrenamiento con sus etiquetas o categorías asociadas según el dominio del problema. El objetivo de estos tipos de algoritmos es aprender la relación (mapeo) que existe entre cada muestra de dato de entrada con su salida (etiqueta o categoría).

En visión computacional, la tarea de clasificación se resuelve prácticamente de forma directa al determinar atributos relevantes de una imagen y asociarlos con su categoría. En la detección de objetos no necesariamente es tan simple, ya que se debe identificar al objeto de interés y luego enmarcarlo para indicar su detección.

Recordemos, por otro lado, que Las redes neuronales artificiales (conocidas como ANN, por sus siglas en inglés), on algoritmos de aprendizaje automático que aprenden de los datos y se especializan en el reconocimiento de patrones inspirados en la estructura y función del cerebro. El aprendizaje profundo (AP) pertenece a la familia de algoritmos ANN y, en la mayoría de los casos, los dos términos se pueden usar indistintamente.

Existen varios elementos que se requieren para lograr el éxito de la implementación de una solución de AP, desde la selección del problema hasta la selección de la mejor infraestructura. Dada la complejidad, es fácil subestimar el rol fundamental que desempeñan los datos en el proceso.





Clasificación de imágenes.

Su objetivo es predecir etiquetas de salida o categorías en función de los datos de entrada, según lo que el modelo haya aprendido de esa relación entrada-salida durante su entrenamiento.

Una tarea de clasificación de imágenes es el reconocimiento de estas, por ejemplo, determinar el tipo de animal que existe en una imagen digitalizada. A partir de una imagen, el sistema determinará (salida) el tipo de animal que está en ella (gato, caballo, etc.) o si no existe animal alguno.

El conjunto de entrada sería una gran colección de imágenes de animales con su categoría asociada. Las imágenes pueden estar en varios formatos (png, jpg, etc.) y puede haber diferencias en las imágenes, independientemente del animal que se incluya, como la escala, el animal puede estar centrado o movido hacia algún extremo, pueden existir otros objetos en la imagen, etc.

Para solucionarlo, contamos con las **redes neuronales convolucionales** (CNN, por sus siglas en inglés) tienen una arquitectura inspirada en el funcionamiento de la corteza visual, que es la responsable de procesar la información visual en el cerebro. La corteza visual primaria (V1) realiza la detección de bordes a partir de la información de entrada en la retina (Woodruff, 2018), y así sucesivamente.



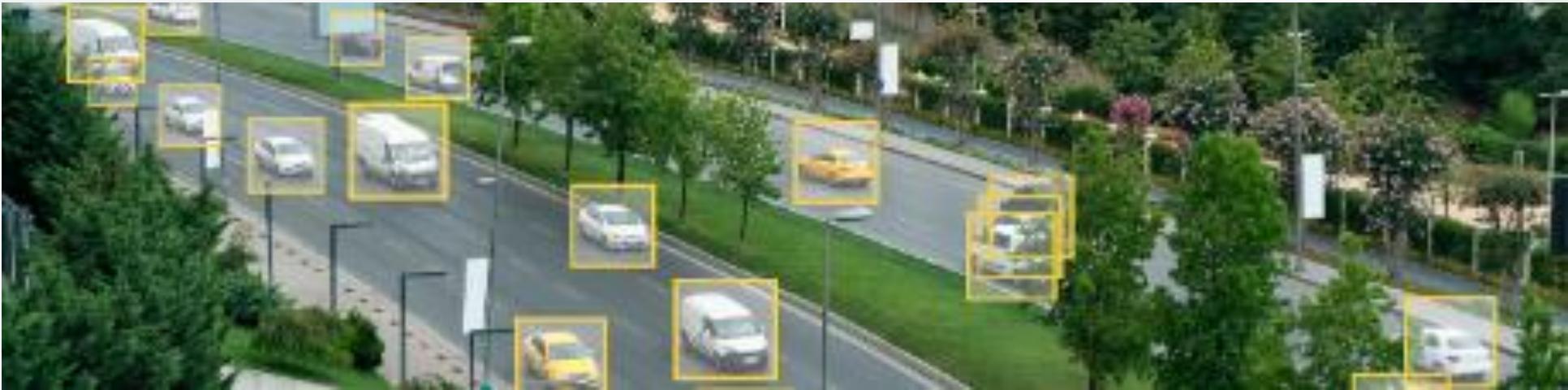


La **convolución** es un término matemático, en el contexto de las CNN es una operación entre matrices. Las capas convolucionales no son otra cosa más que filtros y mapas de imagen, cuyo objetivo es extraer características de la imagen de entrada, preservando la relación espacial entre los píxeles y tienen definidas internamente matrices de pequeña dimensión conocidas como kernels.

Un **filtro es un kernel** en una capa convolucional y puede procesar la imagen de forma eficiente, ya sea difuminándola, haciéndola más nítida o detectando bordes, entre otros.

La estructura de esta red incluye lo siguiente (Khan, Sohail, Zahoor y Qureshi, 2020):

- Una capa convolucional compuesta de **kernels** convolucionales que producen mapas de características de imagen.
- Una capa de **reducción** (pooling) que reduce la cantidad de parámetros al extraer solo las características más comunes.
- Una capa **completamente conectada** que se utiliza para la tarea de clasificación, es decir, predecir la categoría de la imagen.



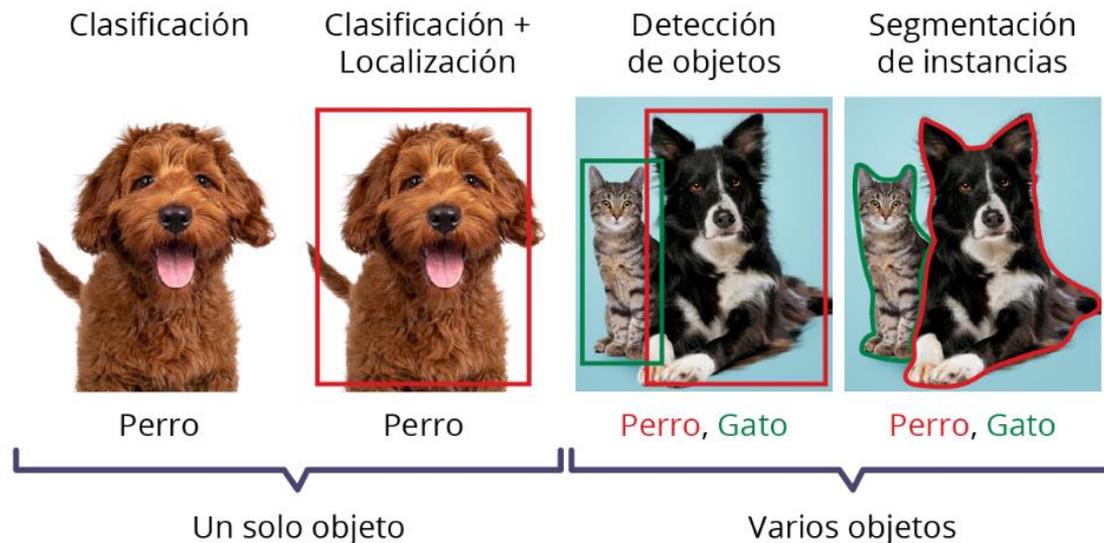


Detección de objetos

La localización de objetos es el nombre que se le da a una tarea de clasificación de objetos con localización. Dada una imagen, se clasifica para cierto objeto que aparece en ella y se determina su ubicación, por lo general, dibujando un recuadro alrededor de este.

La tarea puede tener dos enfoques de solución según el planteamiento del problema: en el primero, se conoce la cantidad de objetos y es un problema cuya solución es directa, utilizando la clasificación y la localización. En el segundo, se desconoce la cantidad de objetos y se convierte en un problema un poco más complejo de detección de objetos.

Ver la figura para clarificar la idea.





Enfoques de solución de la tarea de detección de objetos.

En el **primer enfoque** se conoce la cantidad de objetos y su solución es trivial. Se utiliza una CNN que se entrena no solo para clasificar la imagen, sino para obtener las cuatro coordenadas del recuadro que enmarca al objeto. El problema de localización es un simple problema de regresión.

En un **problema de regresión** se predice un valor continuo, una coordenada, por ejemplo. El recuadro que localiza al objeto se puede representar como una tupla de tamaño cuatro $(x, y, alto, ancho)$ donde (x, y) es la coordenada de la esquina superior izquierda del recuadro, el alto del recuadro y su anchura.

En el segundo enfoque, donde se desconoce el número de objetos a identificar y clasificar, se tiene que considerar que la cantidad de coordenadas que el modelo debe generar no es constante. Si el modelo tiene dos objetos, se necesitan ocho coordenadas; si hay cuatro, se requieren 16 coordenadas, y así sucesivamente.

Una alternativa para dar una solución es usar la propuesta de regiones de interés que se aplica en visión computacional (Reshma y Nath, 2021).





Recopilación de datos

a ruta para la obtención de un modelo predictivo, derivado de cualquier proyecto de aprendizaje profundo, puede dividirse en cuatro etapas:

- 1. Definir y preparar el problema.** ¿con qué finalidad deben recopilarse los datos? e esta forma se identifican las fuentes de los datos a utilizar en la etapa siguiente.
- 2. Recopilar y comprender los datos.** Cada instancia de dato nos ofrece una pequeña muestra de la realidad que representa.
- 3. Procesar y preparar los datos.** El procesamiento le permitirá al modelo obtener, de forma clara, los detalles que requiere durante el entrenamiento. La precisión del modelo dependerá de esta etapa.

Con el conjunto procesado de datos, se construyen tres conjuntos:

- Datos de entrenamiento
- Datos de validación
- Datos de pruebas

- 4. Evaluar algoritmos..** Encontrar el algoritmo que mejor explore la estructura de los datos

Un **modelo matemático** describe las relaciones que existen entre los diferentes aspectos de los datos. Por ejemplo, un modelo que recomienda música puede medir la similitud entre los usuarios (según sus hábitos) y recomendar los mismos artistas a usuarios que han escuchado muchas de las mismas canciones.





Diseño de algoritmo

El aprendizaje profundo es una versión actualizada de las redes neuronales artificiales. En las redes neuronales profundas se construyen redes más grandes y complejas que trabajan con grandes conjuntos de datos etiquetados o no etiquetados, como imágenes, audio, texto o video.

Los **algoritmos más populares** en el AP son los siguientes:

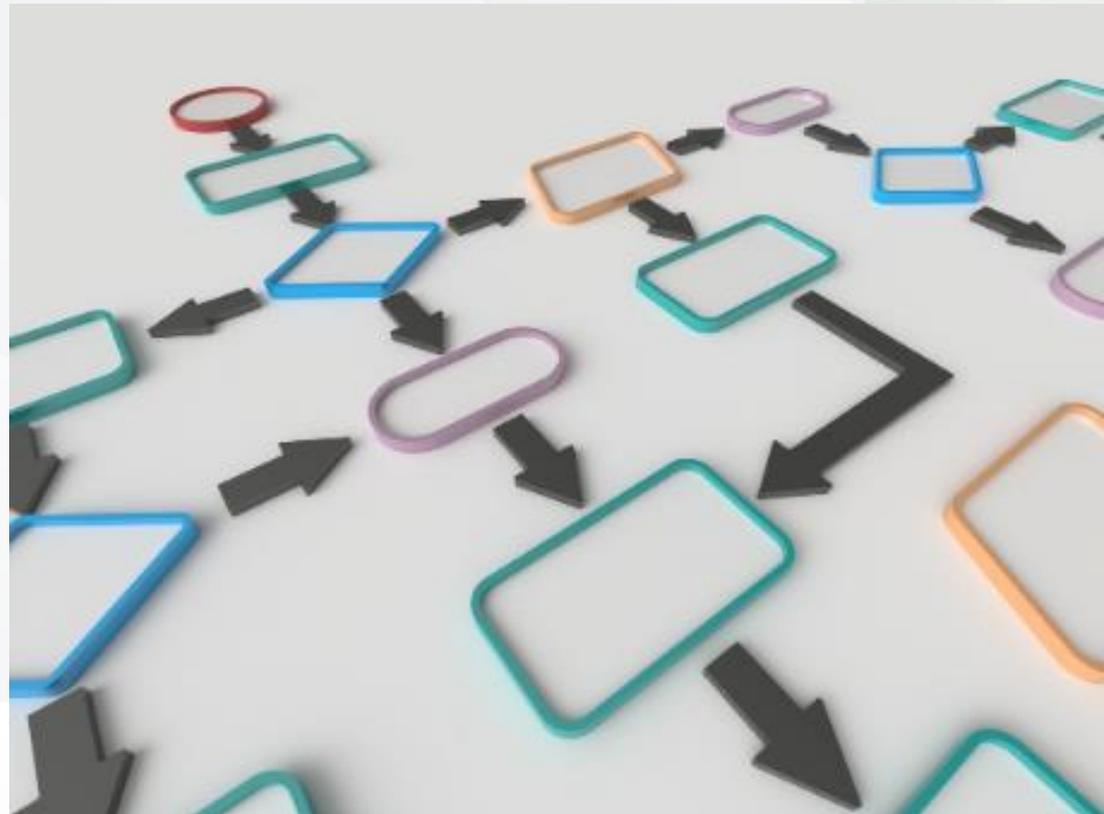
Redes neuronales convolucionales (CNN).

Redes neuronales recurrentes (RNN).

Redes de memoria a corto plazo (LSTM).

Autocodificadores.

Redes profundas de creencias (DBN).





Construir el algoritmo de AP es un proceso que se puede dividir en etapas (Sparr, 2021-a):

1. Comprender el algoritmo a implementar. Tener claro tipo de algoritmo y tipo de tareas
2. Encontrar fuentes de consulta. Recopilar información de libros, artículos, reseñas, páginas web, titulares y/o revistas. Buscar sobre todo claridad en los detalles matemáticos de los algoritmos.
3. Separar el algoritmo en bloques funcionales. Delinear la estructura del algoritmo.
4. Iniciar un ejemplo sencillo. Una especie de prototipo, es decir, codificar e implementar el algoritmo con datos de ejemplo.
5. Validar la implementación del algoritmo. Comparar resultados del diseño con los de la implementación.
6. Documentar el diseño. Para tener una comprensión más profunda, de quien diseñó y de terceros.





Para aprender más, realiza las siguientes actividades:

1. Refuerza el entendimiento de las etapas de un proyecto de aprendizaje profundo para su aplicación en la industria, investigando algunos videos confiables.
2. Practica las tareas de procesamiento de datos para utilizarlas antes del entrenamiento de un algoritmo de aprendizaje profundo.
3. Refuerza el entendimiento de las etapas de un proyecto de aprendizaje profundo para su aplicación en la industria.
4. Practica las tareas de procesamiento de datos para utilizarlas antes del entrenamiento de un algoritmo de aprendizaje profundo.





Es muy importante poner especial atención en el procesamiento y preparación de los datos durante el desarrollo de un proyecto de AP, ya que se realizan tareas que determinan de forma importante el éxito de este, como la preparación y obtención de los conjuntos de datos que se utilizarán durante la etapa de entrenamiento, validación y pruebas del algoritmo de AP. La preparación de datos es particularmente un proceso exploratorio e iterativo, cuyo objetivo es comprender qué representación de los datos ofrecen los mejores resultados.





- Khan, A., Sohail, A., Zahoor, U., y Qureshi, A. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8). Recuperado de <https://link.springer.com/article/10.1007/s10462-020-09825-6>
- Reshma, S., y Nath, P. (2021). Object detection through region proposal based techniques. Recuperado de <https://www.sciencedirect.com/science/article/pii/S2214785321016746#>
- Sparr, M. (2021-a). Demystifying Machine Learning by Building an ML Pipeline (Part 1/2). Recuperado de <https://blog.doit-intl.com/demystifying-machine-learning-by-building-an-ml-pipeline-part-1-43fc30eba242>
- Sparr, M. (2021-b). Demystifying Machine Learning by Building an ML Pipeline (Part 2/2). Recuperado de <https://blog.doit-intl.com/demystifying-machine-learning-by-building-an-ml-pipeline-part-2-dfd9ce786088>

