Cuestionario 2 - Visión por computador

Francisco Javier Sáez Maldonado

Nota.- Las fuentes citadas son enlaces web de donde se ha extraído parte de la información. Puede pulsar sobre ellas para abrirlos en su navegador.

1.Identifique las semejanzas y diferencias entre los problemas de: a) clasificación de imágenes; b) detección de objetos: c) segmentación de imágenes; d) segmentación de instancias.

En la **clasificación de imágenes**, queremos simplemente dada una imagen, asignar una etiqueta a esa imagen. Los siguientes problemas, tendrán diferentes objetivos.

En la **detección de objetos** ,tenemos como objetivo encontrar objetos sobre una imagen, crear una *bounding box* alrededor de esos objetos y predecir la clase de ese objeto, no sabemos de antemano cuántos objetos tendremos en la imagen, puede ser que más de uno, así que podríamos asignar más de una clase, al contrario que en el caso anterior.

En la **segmentación de imágenes**, el problema será asignar *a cada pixel* (cosa que no ocurre en ninguno de los demás casos) una categoría en esa imagen, por lo tanto , no se hará una diferenciación de las instancias de los objetos en la imagen como se tratará de hacer, al contrario que en el caso de la detección de objetos. Podrían aparecer por tanto Al final tendremos la imagen dividida en regiones importantes de la misma.

Por último, podríamos decir que la **segmentación de instancias** es *el problema completo*, pues se trata en este caso de localizar cada entidad en nuestra imagen(como en localización de objetos), realizar su segmentación (es decir, indicar qué píxeles forman parte de esta entidad), pero también queremos clasificarla, pues en este caso, como en la detección de objetos, tendremos múltiples objetos en la misma imagen.

Fuente: cs231-stanford

2.¿Cuál es la técnica de búsqueda estándar para la detección de objetos en una imagen? Identifique pros y contras de la misma e indique posibles soluciones para estos últimos.

La técnica estándar para la detección de objetos es *sliding window*. Esta técnica no es robusta cuando hay pequeños cambios en las formas, o cuando el tamaño de la ventana no es capaz de abarcar el objeto completo. Esto puede solucionarse sin embargo identificando distintas partes del objeto en vez del objeto completo.

También, usando una pirámide con diferentes tamaños de la imagen nos permite identificar con nuestra ventana objetos que podrían haber sido mucho más pequeños o grandes que la plantilla original. *Sliding window* es un algoritmo muy lento para las *CNNs* actuales.

Fuente: cs131-stanford

3.Considere la aproximación que extrae una serie de características en cada píxel de la imagen para decidir si hay contorno o no. Diga si existe algún paralelismo entre la forma de actuar de esta técnica y el algoritmo de Canny. En caso positivo identifique cuales son los elementos comunes y en que se diferencian los distintos.

Existe paralelismo entre ambos algoritmos, pues en cambos casos se intentará obtener la información de los píxeles calculando la intensidad de los gradientes en la imagen para detectar los contornos en la imagen.

La diferencia sin embargo es que en la aproximación (que podría ser *HOG* o *SIFT*), al final se da un vector de características para clasificar la imagen, y en el algoritmo de *Canny* lo que se trata es de hallar solamente los bordes en una imagen.

4.Tanto el descriptor de SIFT como HOG usan el mismo tipo de información de la imagen pero en contextos distintos. Diga en que se parecen y en que son distintos estos descriptores. Explique para que es útil cada uno de ellos.

Se parecen en que ambas calculan los histogramas de gradientes de una imagen, pero se diferencian en que:

- **HOG**, simplemente calcula un histograma de gradientes orientados en toda la imagen
- **SIFT** calcula también un *HOG* pero lo hace tomando divisiones,generalmente 16×16 y luego dividiéndolas en 4×4 donde calcula el *HOG* mencionado. Además, cuando lo ha calculado, utiliza una gaussiana con el tamaño de la mitad de la ventana.

Así, podemos decir que *SIFT* ayuda a describir la importancia de un *punto o región* concreta, mientras que *HOG* es más útil para hacer la clasificación de la imagen concreta.

5.Observando el funcionamiento global de una CNN, identifique que dos procesos fundamentales definen lo que se realiza en un pase hacia delante de una imagen por la red. Asocie las capas que conozca a cada uno de ellos

Los dos procesos fundamentales que se realizan en una *CNN* son:

- Extracción de características de la imagen
- Clasificación de la imagen en función de las características extraídas

A la **extracción de características de la imagen** podemos asociar las capas:

- Capa de convolución (*Conv2D* en keras), que aplican convoluciones a la imagen
- Capa *Relu* , que realiza la función $g(x) = max(0, x)$ a cada pixel de la imagen
- Capa de Normalización en Batch, que normaliza por batches la imagen
- Capa *Pooling*, que realiza una reducción en el tamaño de la imagen

A la capa de **clasificación de la imagen**, que recibe un vector z de características y hace procesamiento de las mismas hasta dar un vector x que tiene en la posición $i-ésima la probabilidad de la imagen de pertenecer a la clase c_i. Esta probabilidad se calcula con la distribución exponencial. Si K es el número de clases,

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

Las capas que podemos encontrar en este proceso son:

- Capas Linear, que hacen operaciones sobre el vector para procesar las características
- Nuevas capas de activación Relu, Dropout o Normalización
- Activación *Softmax*, que calcula la probabilidad final

6.Se ha visto que el aumento de la profundidad de una CNN es un factor muy relevante para la extracción de características en problemas complejos, sin embargo este enfoque añade nuevos problemas. Identifique cuales son y qué soluciones conoce para superarlos.

Al añadir nuevas capas a la red, estamos añadiendo nuevos parámetros al modelo, por lo que probablemente el tiempo de ejecución del entrenamiento sea más largo. Además y de forma más importante, añadir más capas podría llevarnos a **overfitting** en nuestro modelo si tratamos de buscar más parámetros que no sean verdaderamente necesarios, obteniendo así peores porcentajes de acierto en el conjunto de *test*.

7.Existe actualmente alternativas de interés al aumento de la profundidad para el diseño de CNN. En caso afirmativo diga cuál/es y como son.

Las alternativas son diversas. Tratamos en principio de añadir capas que no sean de convolución. Mencionamos algunas de las formas más importantes:

- Añadir capas *Dropout*, que hacen que algunas neuronas dejen de entrenarse según una probabilidad dada *rate*. Esto ayuda a nuestra red a descartar algunas características y evitar así el *overfitting*
- Añadir capas de normalización, como bien podría ser *Batch Normalization*, o normalizaciones L_1 ó L_2
- Aumentar la anchura de la red, añadiendo en las capas convolucionales ya existentes más filtros.
- Si la red ya es profunda, se puede incrementar el número de *skip connections* (conexiones entre capas pero evitando las activaciones no lineales), lo cual mejora el entrenamiento de estas redes profundas (Fuente eminxaq)

8.Considere una aproximación clásica al reconocimiento de escenas en donde extraemos de la imagen un vector de características y lo usamos para decidir la clase de cada imagen. Compare este procedimiento con el uso de una CNN para el mismo problema. ¿Hay conexión entre ambas aproximaciones? En caso afirmativo indique en que parecen y en que son distintas.

Existe conexión entre ambas, pues con una *CNN* lo que estamos haciendo justamente es primero **extraer** las características de la imagen mediante diferentes capas y luego clasificando la imagen mediante una activación *Softmax*. Así, ambos enfoques nos dan el mismo resultado, la clasificación de la imagen.

La **diferencia** entre ellas sería que en el primer enfoque no sabemos cómo estamos extrayendo las características entre las imágenes.

9.¿Cómo evoluciona el campo receptivo de las neuronas de una CNN con la profundidad de la capas? ¿Se solapan los campos receptivos de las distintas neuronas de una misma profundidad? ¿Es este hecho algo positivo o negativo de cara a un mejor funcionamiento?

Cuanto más profunda hacemos nuestra *CNN*, el campo receptivo aumenta, pues cada capa extra incrementa el campo receptivo en *kernel_size* unidades. Además, al hacer *pooling* también hacemos un gran incremento (multiplicativo) de este campo. Estos campos receptivos pueden solaparse en una misma capa, para cubrir toda la imagen.

Además, que estos campos se solapen tiene un efecto positivo, pues hacer los cálculos de forma múltiple sobre los píxeles nos ayuda a obtener mejores características de la imagen.

10.¿Qué operación es central en el proceso de aprendizaje y optmización de una CNN?

Es la operación de activación *Relu*. Esta, dada una entrada x, calcula:

$$g(x) = \begin{cases} 0 & si \ \ x < 0 \\ x & si \ \ x \geq 0 \end{cases}$$

Esta ayuda al aprendizaje de la red, aportando la no linealidad a los modelos que se definen. También acelera el entrenamiento, pues ayuda a calcular de forma más rápida los gradientes. Además, ayuda también a la optimización pues ayuda a solucionar el problema de los gradientes desvanecientes.

11.Compare los modelos de detección de objetos basados en aproximaciones clásicas y los basados en CNN y diga que dos procesos comunes a ambos aproximaciones han sido muy mejorados en los modelos CNN. Indique cómo.

En las aproximaciones clásicas se trabaja con regiones pequeñas de las imágenes y por tanto perdemos generalidad en el proceso. Se tienen por tanto problemas como:

- Una plantilla que recorre la imagen en búsqueda de un objeto, a veces no representa una categoría completa
- Muchos objetos pueden verse de manera diferente según la posición o siendo distintos pero del mismo tipo de objeto

Sin embargo, estos problemas se han **mejorado** mediante las CNN con procesos como:

- Generar y evaluar un gran número de regiones propuestas, que pueden ser más o menos orientadas a nuestras categorías
- Utilizar segmentación jerárquica de imágenes para hallar este tipo de regiones
- Uso de *EdgeBoxes*

Con esto se llega a las *R-CNN*, que hallan primero las regiones relevantes y luego extraen las características de las mismas mediante redes neuronales convolucionales. Se trata de extraer una clasificación para cada *RoI (region of interest)*

12.Es posible construir arquitecturas CNN que sean independientes de las dimensiones de la imagen de entrada. En caso afirmativo diga cómo hacerlo y cómo interpretar la salida.

Sí , es posible. Las capas de convolución pueden tomar inputs de tamaño arbitrario, son las capas *Linear (Dense)* las que necesitan un input fijo. Así, lo que se puede hacer es tratar de hacer las capas *Linear* como una capa de *Convolución*.

Otra técnica posible es **Spatial Pyramid pooling**, que hace *mapas de características* para la imagen completa, y luego hace *sub-imágenes* arbitrarias para crear representaciones de tamaño fijo para poder pasárselo a las capas *Linear* del modelo. Fuente: he-zhang-ren-sun

La **salida** puede interpretarse como una clasificación de cada pixel de la imagen (una imagen segmentada).

13.Suponga que entrenamos una arquitectura Lenet-5 para clasificar imágenes 128x128 de 5 clases distintas. Diga que cambios deberían de hacerse en la arquitectura del modelo para que se capaz de detectar las zonas de la imagen donde aparecen alguno de los objetos con los que fue entrenada.

Como sabemos, teniendo la arquitectura podemos añadir capas delante o detrás de la misma. Como ya tenemos un clasificador, lo que nos hace falta es detectar las regiones potenciales donde podrían estar los objetos que puedan ser clasificados.

Es por ello que lo que tendríamos que hacer es añadir las capas necesarias para crear un nuevo modelo que sea de la forma de una *R-CNN (region proposals+cnn features)*, que primero propone las regiones donde hay objetos identificables y luego los identifica.

14.Argumente por qué la transformación de un tensor de dimensiones 128x32x32 en otro de dimensiones 256x16x16, usando una convolución 3x3 con stride=2, tiene sentido que pueda ser aproximada por una secuencia de tres convoluciones: convolución 1x1 + convolución 3x3 + convoluión 1x1. Diga también qué papel juegan cada una de las tres convoluciones.

Lo que hará la primera convolución (1×1) será seleccionar sobre la imagen las características importantes, haciendo la imagen menos profunda y que tengamos que hacer menos cálculos en el filtro (3×3).

Ahora, el filtro (3×3) hará lo mismo que haría una convolución normal (3×3), tratar de extraer ciertas características.

Con la última convolución (1×1), lo que se hará es hacer la salida del tamaño que nos interesa, es decir, como si hubiésemos hecho una convolución (3×3) con $stride = 2$.

Es por ello que podríamos obtener resultados similares haciendo un número inferior de cálculos y por tanto habiendo mejorado la eficiencia de la convolución inicial de $stride = 2$.

15.Identifique una propiedad técnica de los modelos CNN que permite pensar que podrían llegar a aproximar con precisión las características del modelo de visión humano, y que sin ella eso no sería posible. Explique bien su argumento.

Una propiedad técnica importante en las *CNN* es la jerarquía de las capas, que nos permite conforme avanzamos en profundidad extraer más características de una imagen. El modelo de visión humano funciona de la misma manera, pues se procesan de forma jerárquica en nuestro *córtex* diversas características de los objetos que visualizamos , obteniendo cada vez características más abstractas del objeto que visualizamos.

En nuestro entrenamiento, pretendemos también obtener características lo más abstractas posibles, para lo cual usamos un *orden o jerarquía* en las capas, de forma que consigamos un extractor de características que obtenga estas características de forma lo más genérica posible, sin llegar a hacer overfitting.