UNIVERSIDAD
DE GRANADA

Visión por Computador
grado en ingeniería informática

Trabajo 2

Cuestiones de teoría

Autor
Vladislav Nikolov Vasilev

Rama
Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Curso 2019-2020

Índice

Ejercicio 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.

Solución

Ejercicio 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.

Solución

Ejercicio 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.

Solución

Ejercicio 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.

Solución

Ejercicio 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.

Solución

El primer proceso fundamental es la **extracción de características**. Mediante este proceso se puede extraer información relevante sobre la imagen de entrada. En este proceso intervienen toda una serie de capas, como las capas convolucionales, las cuáles realizan transformaciones sobre la imagen de entrada; las capas de activación, las cuáles introducen alguna función no lineal sobre las salidas de las capas convoluciones, como por ejemplo la función `ReLU`; y las capas de *pooling*, las cuáles realizan una reducción o aumento del tamaño de la imagen. También se pueden utilizar otras capas en este proceso las cuáles sirven para regularizar el modelo, como por ejemplo las capas de `Dropout` o de `Batch Normalization`. De esta forma, se puede llegar a evitar el sobreajuste que se pueda producir en el modelo.

El segundo proceso funtamental es la **predicción**, la cuál utiliza la información extraída en el proceso anterior para proporcionar algún tipo de información de salida. Por ejemplo, se puede predecir a qué clase pertenece una imagen dada. En esta parte se utilizan normalmente capas totalmente conectadas con una función de activación determinada en la última capa. En el ejemplo de clasificación anterior se utilizaría la función `softmax`, la cuál da un vector de probabilidades para cada una de las clases.

Ejercicio 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.

Solución

Al aumentar la profundidad de la red, se pueden extraer mejores características de la imagen, y por ende, se pueden aprender funciones más complejas. Sin embargo, existe un problema con este enfoque, y es que llega un punto en el que la función que se aprende se empieza a pegar demasiado a los datos de entrenamiento, y por tanto se produce sobreajuste. Para evitarlo, se pueden introducir capas de regularización,

como por ejemplo `Batch Normalization`, `Dropout` o haciendo un *early-stopping*, de forma que se pare de entrenar antes de que se produzca sobreajuste.

Otro problema que nos encontramos al aumentar la profundidad es que llega un punto en el que el error propagado por el algoritmo de `Back Propagation` es 0, con lo cuál no llega nada a las primeras capas y no se produce un ajuste de los pesos en función del gradiente. Este problema tiene distintas soluciones, como por ejemplo modificar la red de forma la arquitectura no sea secuencial, haciendo que una capa no esté conectada solamente con la siguiente, si no con otras, como por ejemplo se hace en la red residual `ResNet` [1]; o también introduciendo más de un clasificador en la red, como por ejemplo en `GoogLeNet` [2], de forma que el gradiente no pierda su intensidad.

Ejercicio 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.

Solución

Ejercicio 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.

Solución

Ejercicio 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?

Solución

Ejercicio 10

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

Solución

Ejercicio 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.

Solución

Ejercicio 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.

Solución

Ejercicio 13

Suponga que entrenamos una arquitectura Lenet-5 para clasificar imágenes 128×128 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.

Solución

Ejercicio 14

Argumente por qué la transformación de un tensor de dimensiones $128 \times 32 \times 32$ **en otro de dimensiones** $256 \times 16 \times 16$**, usando una convolución** 3×3 **con stride=2, tiene sentido que pueda ser aproximada por una secuencia de tres convoluciones: convolución** 1×1 **+ convolución** 3×3 **+ convoluión** 1×1**. Diga también qué papel juegan cada una de las tres convoluciones.**

Solución

Esta transformación tiene sentido ya que las convoluciones 1×1 sirven para aumentar/reducir la profundidad sin modificar las otras dimensiones del tensor, mientras que la convolución 3×3 del segundo modelo va a hacer exactamente lo mismo que la del modelo original: extraer característsticas y reducir la anchura y la altura del tensor a la mitad, ya que en ambos casos se utiliza un `stride = 2`. Por tanto, lo único nuevo que se introduce en la segunda transformación es la reducción y el aumento de la profundidad del tensor, conservando convolución central (la 3×3). Este segundo enfoque, aunque en un principio parezca que no, tiene una serie de beneficios.

Por una parte, se consigue un **aumento de velocidad** ya que el número de operaciones a realizar es menor. Para ello, siguiendo la línea de lo dicho anteriormente, la primera convolución 1×1 reduciría la profundidad del tensor, compactándolo. La segunda convolución, una 3×3 con `stride = 2` se aplicía sobre el resultado anterior, reduciendo por tanto la anchura y la altura del tensor a la mitad, pero conservando la profundiad, y extrayendo características en el proceso. Finalmente, la tercera convolución 1×1 aumentaría la profundidad del tensor a 256, el tamaño de salida. De esta forma, tendríamos una arquitectura que consiste en reducir la dimensionalidad, extraer características y aumentar la dimensionalidad.

Para verlo más claro, vamos a ver las operaciones que se realizan en cada caso de forma gráfica. Vamos a suponer que en el segundo caso, al reducir la dimensionalidad con la primera convolución 1×1 se pasa a 64 canales, ya que no se especifica nada sobre el número de canales de salida en el enunciado.

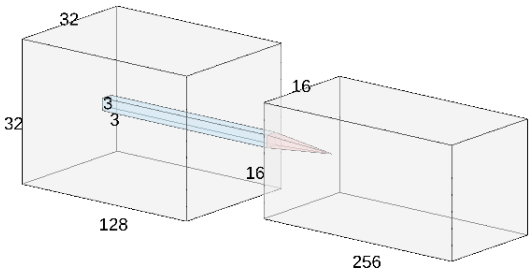

Figura 1: Transformación utilizando una convolución 3×3 con `stride = 2`.

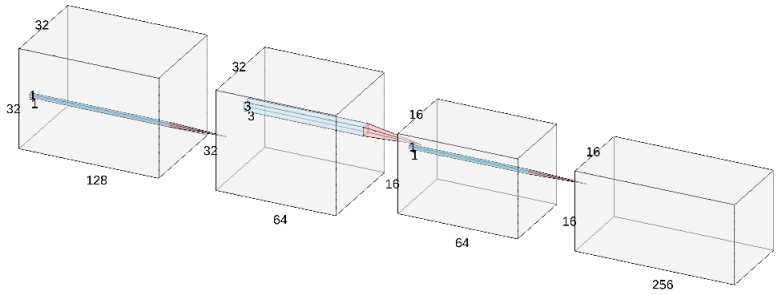

Figura 2: Transformación utilizando tres convoluciones.

Se puede ver que al final el tensor de salida es el mismo, solo que en un caso se realizan más tranformaciones. Sin embargo, si nos paramos a analizar el número de operaciones para cada caso, se obtienen los siguientes resultados:

- Para el caso en el que solo se hace una convolución, se hacen $128 \times 256 \times 3 \times 3$ operaciones, lo que son unas `295K` operaciones en total aproximadamente.
- Para el caso en el que se hacen 3 convoluciones, tenemos que en la primera convolución se hacen $128 \times 64 \times 1 \times 1$ operaciones, en la segunda $64 \times 64 \times 3 \times 3$ y en la tercera se hacen $64 \times 256 \times 1$. Este número de operaciones on `8K`, `36K` y `16K` operaciones en cada caso, lo cuál son en total aproximadamente unas `60K` operaciones.

Por tanto, vemos que en el segundo caso se hacen menos operaciones, y por consiguiente, será una arquitectura más rápida que la del primer caso.

Por otra, se consigue un **aumento de la no linealidad**. Esto se debe a que entre cada capa de convolución se puede insertar una función de activación, la cuál introduce no linealidad. Este aumento es beneficioso, ya que se puede conseguir una mejor aproximación a la función real que se quiere aprender, la cuál es muy difícil que sea lineal (una función lineal es demasiado simple como para aprender algo así con una red convolucional, para eso podríamos resolver el problema con un clasificador lineal). En el segundo caso, podemos introducir hasta tres capas de activación (las cuáles introducen la no linealidad), mientras que en la primera solo tenemos una activación al final del proceso, con lo cuál se consigue una aproximación "peor" (que no tiene por qué ser mala) de la función real. Sin embargo, es esta no linealidad la que hace que los resultados no sean exactamente los mismos en los dos casos, si no que sean aproximados.

Ejercicio 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.

Solución

Referencias

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.