

Artificial neural networks

Convolutional neural networks

Traffic Sign Recognition

Adrián Vázquez Barrera

Polytechnic University of Valencia

January 2022

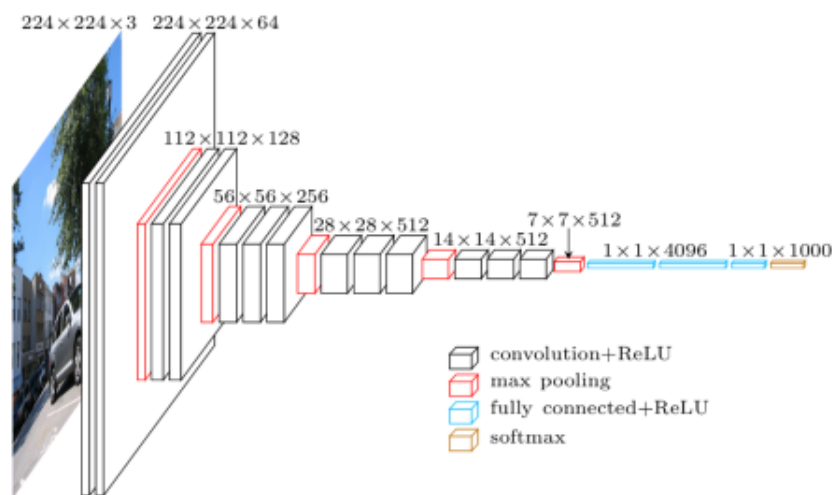
Contexto

Se ha desarrollado una red neuronal convolucional especializada en reconocer 42 señales alemanas de tráfico. Para ello, se ha utilizado el conjunto de datos GTSRB alojado en la plataforma Kaggle. Este dataset cuenta con más de 50 mil imágenes, con sub-conjuntos de entrenamiento y test previamente separados. Lo que es de agradecer, puesto que no encontramos el mismo número de ejemplos para todas las clases existentes y ahorra trabajo de preprocesamiento.

Adicionalmente, se ha enfocado el desarrollo de la práctica a una visión de producto, más que investigación. Es por ello que, además del estudio y análisis del estado del arte para esta tarea. Se ha implementado un contenedor Docker que lee un modelo pre-entrenado y automáticamente sirve de reconocedor de imágenes a través de un servidor web. Esto permitirá acoplar el modelo a cualquier aplicación existente con independencia de las tecnologías (o versiones existentes) empleadas.

Diseño de la red

La estructura de la red se fundamenta en la popular arquitectura VGG (Visual Geometry Group) que se basa en una sucesión profunda de capas convolucionales. La arquitectura VGG es la base de modelos innovadores de reconocimiento de objetos, superando los límites establecidos en muchas tareas y conjuntos de datos más allá de ImageNet.



Solo con aplicar esta arquitectura no es suficiente para obtener una tasa de error de alrededor del 2% que sería lo aceptable. Adicionalmente, se han implementado algoritmos de Mix-Up y Cut-Out. Este último es una función acoplada al Data-generator que, dada una imagen, elimina fragmentos de esta. Lo que obliga a la red a aprender exclusivamente aquellos elementos fundamentales de las muestras en el entrenamiento. Por otro lado, con Mix-Up conseguimos fusionar dos muestras del conjunto de entrenamiento para construir una nueva imagen modificando el canal alpha. Con ello, estableceremos deliberadamente la preponderancia de una imagen sobre otra en un 10%. Lo que añade ruido en el entrenamiento, evitando el sobre-aprendizaje de ciertos atributos y ayuda a mejorar la red cuando haya confusión entre dos clases.

Implementación de la red

Tal y como se mencionó en el apartado anterior, la arquitectura de la red utilizada corresponda a una VGG. El número de epochs se ha establecido en 115 con un batch size de 1024. El factor de aprendizaje se encuentra originalmente establecido a 0.001, aunque durante el entrenamiento se hace uso de la técnica "Learning Rate Annealing" con el siguiente planificador:

$$scheduler(epoch) = learning_rate \cdot 2^{-\lfloor \frac{epoch}{20} \rfloor}$$

Se ha apostado por el optimizador Adam, y los siguientes valores para el Data Augmentation:

| Data Augmentation | |
|--------------------|-------------------|
| Rotation range | 10 degrees |
| Zoom range | 0.15 |
| Width shift range | 0.1 |
| Height shift range | 0.1 |
| Shear range | 0.15 |
| Horizontal flip | True |
| Vertical flip | False |
| Cut-Out | Area $\in (0, 1)$ |
| Mix-Up | 0.1 |

Al mismo tiempo, la red guardará una copia del modelo cuando exista un epoch que mejore la tasa de acierto con respecto al conjunto de test.

Despliegue de la red

El despliegue del sistema se ha llevado a cabo utilizando un contenedor docker a partir de una imagen vanilla de python 3.9.7, a la que se ha añadido TensorFlow 2.5.0, Numpy 1.19.5 y OpenCV. La aplicación además emplea las herramientas del lenguaje para habilitar un servidor HTTP que acepta peticiones POST.

Para realizar una predicción, será necesario enviar una petición web a la dirección IP de la máquina y adjuntar una imagen en formato PNG con la señal de tráfico a reconocer.

Bibliografia

1. GTSRB - German Traffic Sign Recognition Benchmark - Multi-class, single-image classification challenge.
<https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>
2. VGG Very Deep Convolutional Networks (VGGNet) - What you need to know.
<https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks>
3. MixUp augmentation for image classification.
<https://keras.io/examples/vision/mixup>
4. Cutout regularization for CNNs.
<https://medium.com/@ombelinelag/cutout-regularization-for-cnns-62670d86bc33>