Master universitario en Inteligencia Artificial

Alberto Bausá Cano

Comparativa de redes convolucionales para clasificación de imágenes en visión artificial

unir LA UNIVERSIDAD EN INTERNET

Universidad Internacional de La Rioja

Índice

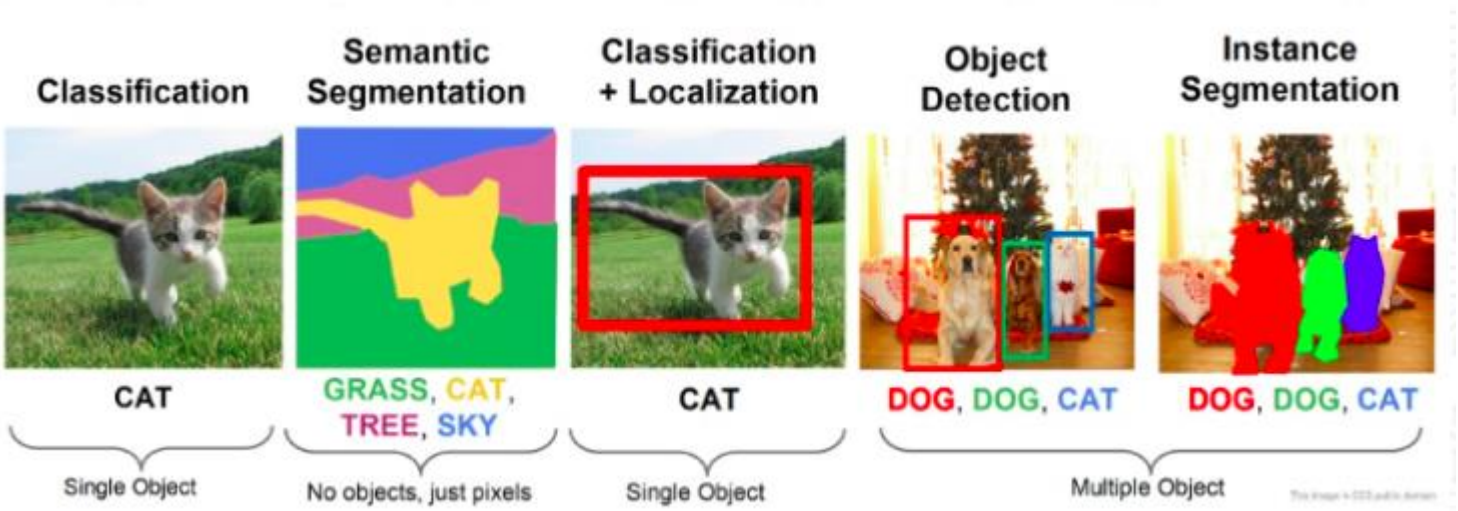
Introducción

- En la IA actual, dos de las principales vías de investigación han sido:
- La Visión Artificial (VA), con multitud de aplicaciones (diagnósticos médicos, detección de anomalías, conducción autónoma…).
- El Aprendizaje Profundo (AP), seno de una de las herramientas tecnológicas más utilizadas hoy en día, las Redes Neuronales (RN).
- De la aplicación del AP a la VA ha surgido, especialmente durante la última década, un tipo concreto de redes conocidas cono Redes Neuronales Convolucionales (RNC), especializadas en tareas de VA.
- A raíz de ello, se ha vivido una reciente proliferación, tanto en la variedad de arquitecturas RNC, como en la disponibilidad de datasets públicos de calidad para entrenar y evaluar tales modelos.
- Esto pone de relevancia la magnitud e importancia que tienen estas técnicas, así como la enorme diversidad de alternativas existentes.

Contexto y Estado del Arte (I)

- En la actualidad, algunas de las tareas o problemáticas más comunes dentro de la VA son las que se pueden apreciar en la Figura.

- Destacan la **clasificación** de imágenes, la **segmentación semántica** (por colores), la **localización**, la **detección de múltiples objetos**, o la **segmentación de instancias** (diferentes objetos o clases).



Fuente: Fei-Fei Li & Justin Johnson & Serena Yeung (2017)

Contexto y Estado del Arte (II)

- Por otro lado, como ejemplos más representativos de arquitecturas RNC de la última década, todas ellas partícipes de competiciones como el ILSVRC (Russakovsky et al., 2015), podemos encontrar:

- **AlexNet**, considerada la primera RNC moderna, fue pionera en ganar la mencionada competición, demostrando su potencial ante problemas como la clasificación de imágenes o la detección de objetos.

- **GoogLeNet**, ganadora en 2014, introducía la interesante novedad de los módulos *Inception*, los cuales permitían reducir la dimensionalidad de los datos, así como realizar varias convoluciones en paralelo.

- **ResNet**, quien se llevó la edición de 2015, presentaba los bloques residuales, los cuales servían como atajos entre capas, permitiendo aumentar exponencialmente la profundidad de la red (hasta 150 niveles), sin que ello supusiera una penalización de rendimiento.

Contexto y Estado del Arte (III)

- Por último, cabe mencionar algunos de los datasets más representativos y empleados de forma reciente en el campo.
- **MNIST**, considerado el “hola mundo” del aprendizaje automático.
- **CIFAR 10 & 100**, con 10 y 100 clases de objetos respectivamente, bases sólidas para evaluar modelos en clasificación de imágenes.
- **Google Open Images**, de propósito general, siendo uno de los más extensos, con cerca de 10 millones de imágenes y sus anotaciones.
- Todas estas alternativas forman parte de un amplio abanico en VA, lo cual, a su vez, pone de manifiesto la importancia de conocer las peculiaridades de cada opción, así como sus ventajas y desventajas.
- De esta manera, surge la idea de desarrollar una comparativa entre algunas de estas soluciones, centrando el foco en la clasificación de imágenes, con el fin de proporcionar indicadores fiables y empíricos.

Objetivos y metodología (I)

- Es así como llegamos al objetivo principal del presente estudio.

Realizar un análisis comparativo que evalúe el rendimiento ofrecido por diferentes arquitecturas de redes centradas en tareas de VA

- De él, se desglosan una serie de objetivos específicos:

- 1. Seleccionar los modelos, datasets y librerías más adecuados.

- 2. Encontrar las mejores implementaciones disponibles para las alternativas anteriormente elegidas.

- 3. Desarrollar el análisis comparativo, ejecutando y evaluando los diferentes modelos, y recolectando un compendio de métricas.

- 4. Debatir los resultados, reflejados en las métricas, tratando de interpretarlas, darles un significado, y extraer conclusiones relevantes.

Objetivos y metodología (II)

- Para alcanzar tanto el objetivo general como los específicos, se plantea la metodología expuesta a continuación:

- **Paso 1**. Proceso de selección sobre el conjunto de frameworks, arquitecturas y datasets previamente descubiertos o estudiados durante el análisis del estado del arte, justificando cada decisión.

- **Paso 2**. Construcción y desarrollo de las diferentes arquitecturas, así como ajuste de las técnicas de preprocesado necesarias en el dataset.

- **Paso 3**. Ejecución de los modelos y recolección de métricas.

- **Paso 4**. Análisis de los resultados, estudiando el comportamiento exhibido por cada uno de los modelos y conjuntos de técnicas de entrenamiento empleadas.

Desarrollo de la comparativa (I)

- Para poder evaluar las arquitecturas seleccionadas, se ha elegido **CIFAR-10** como dataset. Está especializado en la clasificación de imágenes (10 clases), es suficientemente sencillo como para ser manejable, y a su vez ofrece la complejidad requerida para poner realmente a prueba los diferentes modelos.

- En cuanto a arquitecturas, se han seguido dos filosofías:
- Por un lado, una red ***custom-cnn***, entrenada desde cero, red convolucional sencilla de seis capas y 300mil parámetros entrenables. Se ha empleado *data augmentation* para ampliar el conjunto de datos de partida, añadiendo diversas variaciones a los ejemplos.
- Además, se usa una tasa de aprendizaje adaptativa, mediante un planificador, que modifica el impacto de dicha tasa a lo largo del entrenamiento: más alta al principio, para acelerar la convergencia, y más baja al final, para afinar los resultados.

- Por otro lado, se han utilizado 2 modelos de **redes preentrenadas** (*ImageNet*), aplicando una técnica conocida como *transfer learning*.

Desarrollo de la comparativa (II)

- En primer lugar, tenemos el modelo ***vgg16*** (Simonyan & Zisserman, 2014). De él, se conservan los 3 primeros bloques, y a partir de ahí, se añaden capas densas y de normalización. Además, la base reutilizada se congela, quedándonos con solo 134mil parámetros entrenables, correspondientes a las nuevas capas.

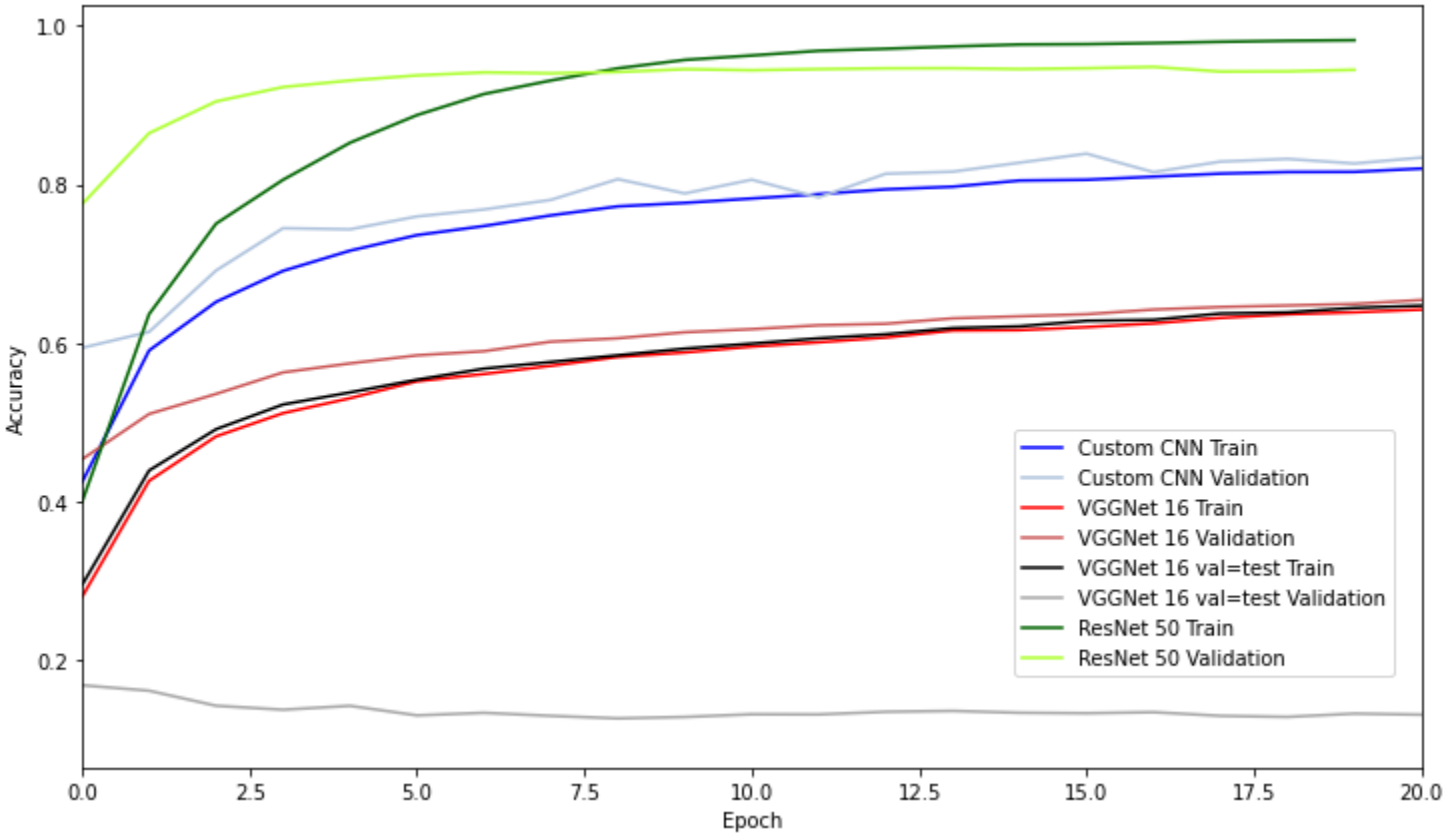
- Se utiliza de nuevo *data augmentation*, así como una tasa de aprendizaje diferencial, dado que para las capas congeladas esa tasa es cero, mientras que para el resto del modelo se usa 4e-4.

- En segundo lugar, tenemos la arquitectura ***resnet50*** (He et al., 2015), con un *dropout* del 50% después de cada capa densa. Asimismo, ahora no se congela ninguna parte del modelo, por lo que, partiendo de los valores ya adquiridos por ese modelo para *ImageNet*, se reentrena en *Cifar-10*, junto con la nueva parte de la arquitectura.

- Por último, se enumeran brevemente las métricas empleadas: *accuracy & loss* (train, validation y test), tiempo de ejecución y número de épocas, así como consumo de memoria y espacio en disco.

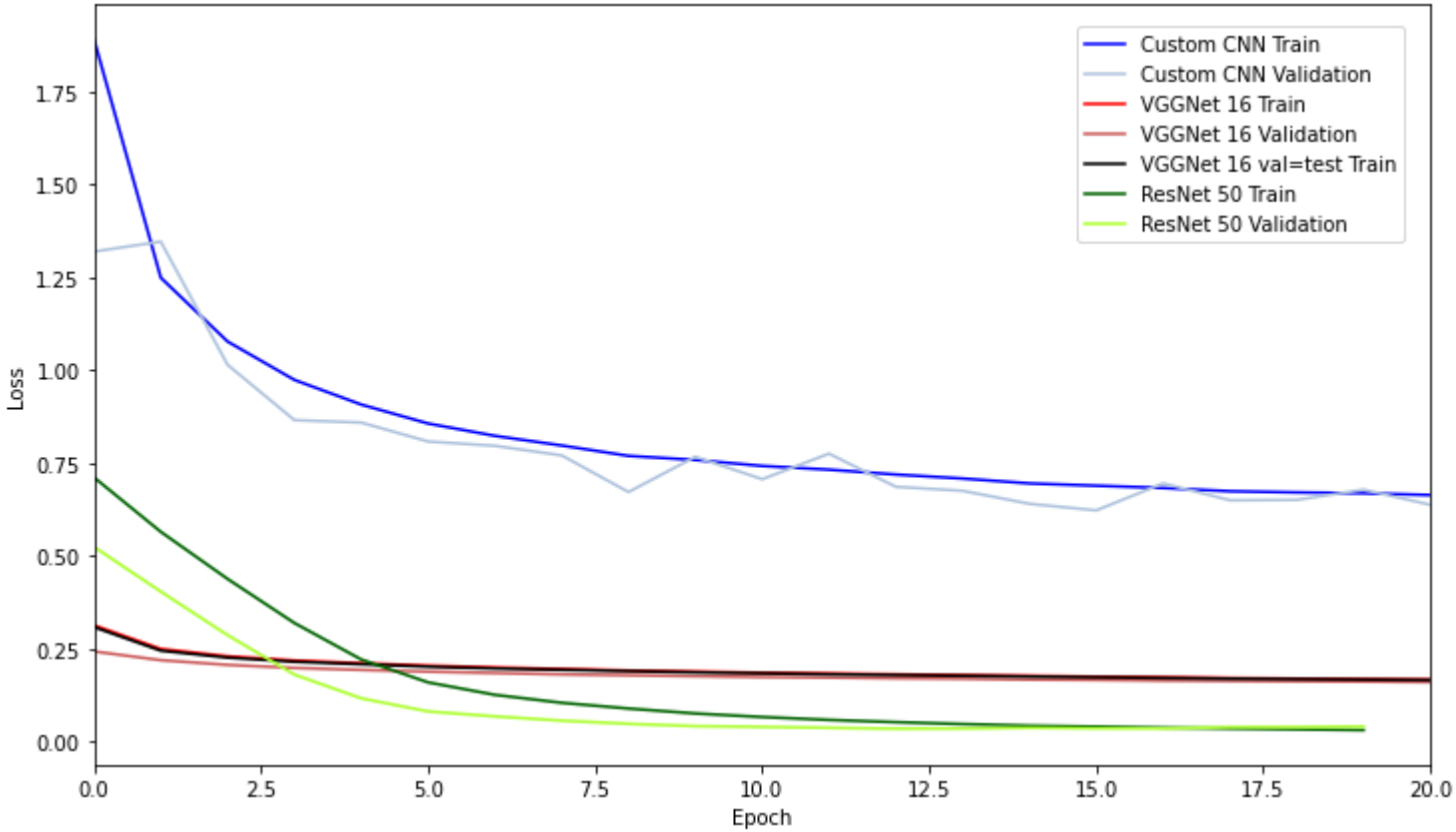
Análisis de resultados (I)

- Se comienza exponiendo dos Figuras con los resultados globales.



Accuracy (precisión) global
Fuente: elaboración propia (2021)

Análisis de resultados (II)



Loss (pérdida) global
Fuente: elaboración propia (2021)

Análisis de resultados (III)

- Como podemos observar en las Figuras anteriores, limitadas a un rango temporal de 20 épocas de ejecución, tenemos los resultados obtenidos, expresados mediante métricas de *accuracy* y *loss* para el entrenamiento y la validación, de cada uno de los modelos: *custom-cnn*, *vgg16* (con dos y tres conjuntos de datos) y *resnet50*.
- Vemos como la que peores resultados cosecha es sin duda la arquitectura *vgg16*, tanto cuando usa 2 conjuntos de datos (uno para train, y otro para val./test), como cuando usa 3 conjuntos (excluyentes para train, val. y test), en sendos casos apenas llega al 80% de *accuracy* en train, con resultados más pobres en validación.
- Le sigue el modelo *custom-cnn*, con una *accuracy* del 89% aproximadamente, teniendo por delante, en primer lugar, la arquitectura *resnet50*, con un sobresaliente 98%.
- Se puede ver como los resultados para la *loss* en cada ejecución, son proporcionalmente inversos a la *accuracy* lograda, teniendo, por orden, la menor pérdida *resnet50*, salvo para *custom-cnn.*

Análisis de resultados (IV)

- De estos datos, se puede inferir que la decisión de congelar totalmente la parte reutilizada en el caso de *vgg16*, no fue acertada, puesto que no le permitió adaptar sus parámetros a las casuísticas del nuevo dataset *Cifar-10*, cosa que si pudo hacer *resnet50*, pues consiguió ajustar sus pesos de tal forma que alcanzase una notable precisión en los nuevos ejemplos sobre los que se entrenaba.

- Por otro lado, también cabe resaltar el contraste entre una red sencilla (*custom-cnn)* con 300mil parámetros, y una red gigantesca (*resnet50)* con cerca de 40 millones entrenables, siendo mucho más profunda.
- De esta forma, mientras que la segunda logra un excelso 98% después de 4h, la primera obtiene un respetable 89%, en la mitad de tiempo, tan solo 2 horas de entrenamiento. Aquí se resalta la importancia de que, si bien llegado a un punto, aumentar el tamaño de la red no significa una mejora, si es necesario una mínima profundidad que le permita aprender las abstracciones correctas de los datos.

- Por último, poner de relevancia que si bien el *transfer learning* demuestra ser una técnica eficiente, hay que saber aplicarla bien.

Conclusiones y trabajos futuros (I)

- A lo largo del presente trabajo se ha realizado una comparativa de diferentes arquitecturas convolucionales, con el propósito de conocer cuáles de estos modelos permiten alcanzar mejores resultados.

- Para ello, nos hemos guiado por una serie de objetivos, a recordar:

  (1) Seleccionar los modelos, datasets y librerías más adecuados.

- Se comenzó con una selección de entre el conjunto de alternativas descubiertas en el estado del arte, justificando cada elección.

  (2) Encontrar las mejores implementaciones disponibles para las alternativas de arquitecturas elegidas.

- Una vez realizada la selección, se hizo una revisión más profunda a nivel técnico, para elegir qué implementación por cada arquitectura podría ser la óptima, así como qué técnicas para el preprocesado de los datos y el entrenamiento eran las más adecuadas en cada caso.

Conclusiones y trabajos futuros (II)

(3) Desarrollar el análisis comparativo, ejecutando y evaluando los diferentes modelos, y recolectando un compendio de métricas.

- Una vez adquirido el conocimiento teórico y técnico necesario, se desarrolló la comparativa, ejecutando de manera ordenada cada arquitectura y configuración para su entrenamiento, así como recolectando una serie de métricas.

(4) Debatir los resultados, reflejados en las métricas, tratando de interpretarlas, darles un significado, y extraer conclusiones relevantes.

- Por último, se concluyó con una interpretación de las métricas obtenidas como producto de la comparativa, analizando punto por punto mediante gráficas el comportamiento de cada modelo.

- Por tales motivos, se creen satisfechos, tanto el objetivo general del estudio, como cada uno de los objetivos específicos expuestos anteriormente. Siendo el fruto de dicho trabajo el conjunto de ideas y conclusiones finales acerca de los resultados.

Conclusiones y trabajos futuros (III)

- Si bien se consideran superados los objetivos de partida, quedan algunas cuestiones pendientes, problemas y oportunidades identificadas durante el desarrollo, que permitirían mejores resultados.
- En primer lugar, dada la gran variedad de alternativas existente, el estudio se presta a su escalabilidad en varias direcciones. Se podrían explorar paradigmas diferentes al convolucional (Redes Generativas Antagónicas, Redes de Cápsulas). Por otro lado, se podrían emplear nuevos datasets, como CIFAR 100, con 100 clases en lugar de 10.
- En relación a los resultados obtenidos, se podrían realizar mejoras en el modelo *vgg16*, analizando las causas de su pobre rendimiento, y comenzando por darle mayor capacidad de aprendizaje aumentando su profundidad, así como probando alternativas de entrenamiento.
- Por último, una solución interesante, dado el contexto del estudio, hubiera sido el desarrollo de una herramienta para comparar, de forma ágil y cómoda, diferentes arquitecturas y datasets, similar a TensorFlow Playground, pero enfocada en redes convolucionales.

Bibliografía

Fei-Fei Li & Justin Johnson & Serena Yeung. (2017). Lecture 11: Detection and Segmentation. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. http://arxiv.org/abs/1409.0575

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs]. http://arxiv.org/abs/1409.1556

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. http://arxiv.org/abs/1512.03385

LA UNIVERSIDAD
EN INTERNET

Muchas gracias

www.unir.net