Deep Learning

Sesión 5 - Resumen

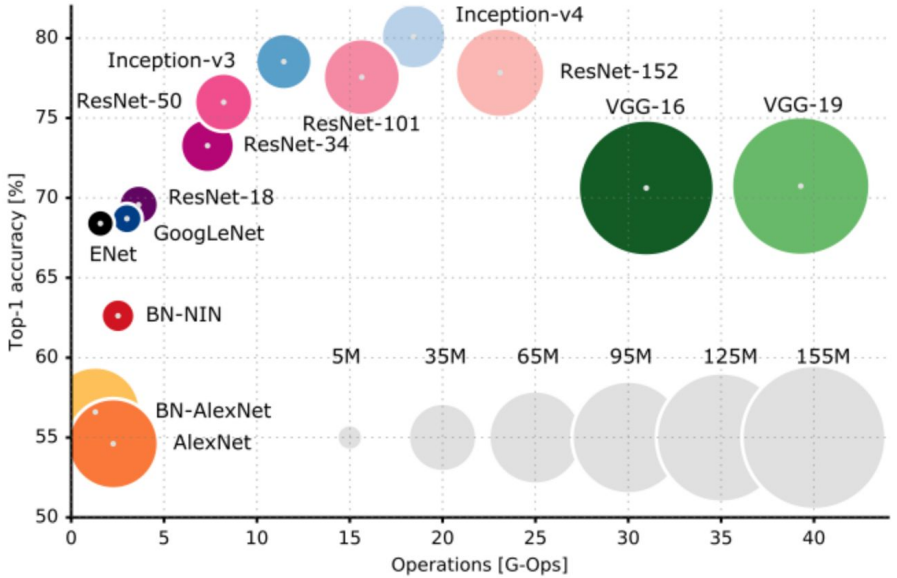
De un vistazo

- Arquitecturas de CNNs más comunes
- Extensiones al Gradient Descent
- Visualización de mapas de activación y filtros
- Transfer Learning
- Fine Tuning
- Data Augmentation

Arquitecturas más comunes

Existen arquitecturas predefinidas que nos evitan el dolor de cabeza de elegirla nosotros. Están más que probadas y se conocen sus bondades/defectos:

- VGG
- ResNet
- Inception
- Xception
- SqueezeNet
- MobileNet
- etc...



Extensiones al Gradient Descent

Además del Descenso del gradiente tradicional (Vanila Gradient Descent) existen mejoras que han ido surgiendo para mejorar la velocidad de convergencia:

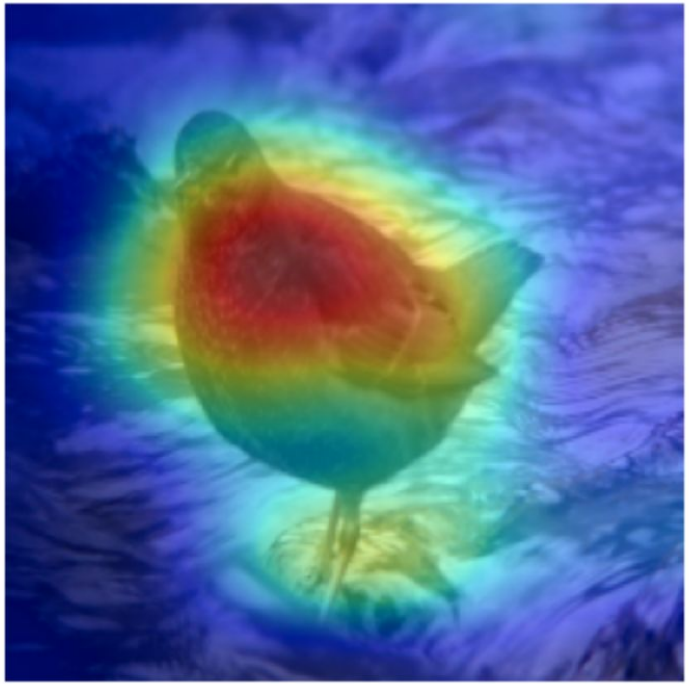
- **Momentum**: aprovechamos la “inercia” para avanzar más cuando estamos más seguros
- **Nesterov momentum**: muy similar al momentum
- **AdaGrad**: adapta el learning rate de acuerdo a cómo de “comunes” sean las features
- **RMSprop**: muy similar a AdaGrad, pero soluciona un problema que hace que AdaGrad acabe usando learning rates muy bajos que previenen el aprendizaje
- **Adam**: mejora de RMSprop que añade momentum

Visualización de mapas de activación y filtros

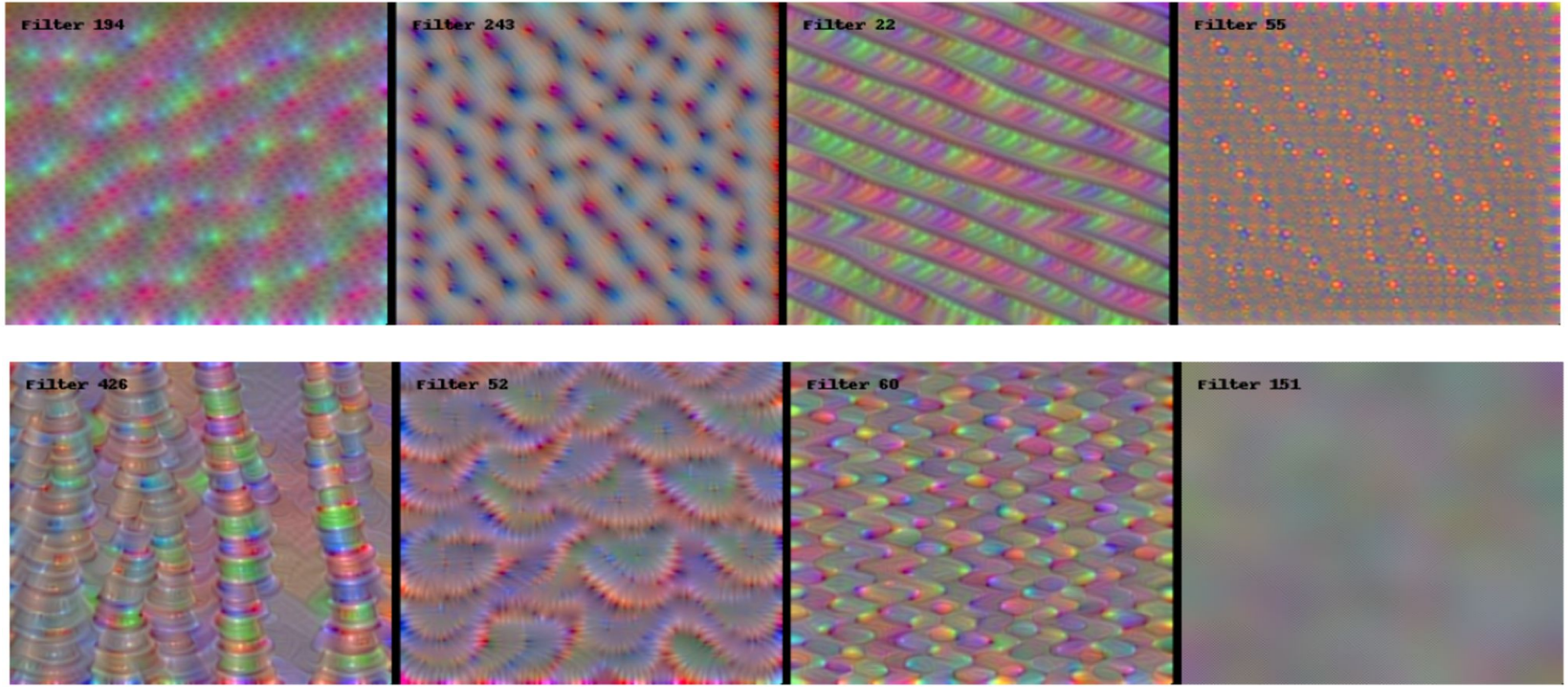
Para conseguir una mayor interpretabilidad de nuestros modelos podemos:

- Visualizar los **mapas de activación** de las capas que deseemos
- Visualizar los **filtros** (o kernels, o pesos) de las capas que deseemos
- Visualizar qué **imagen** es la que consigue la **mayor activación** para una determinada clase, es decir, la imagen “representante” de dicha clase

Visualización de los mapas de activación

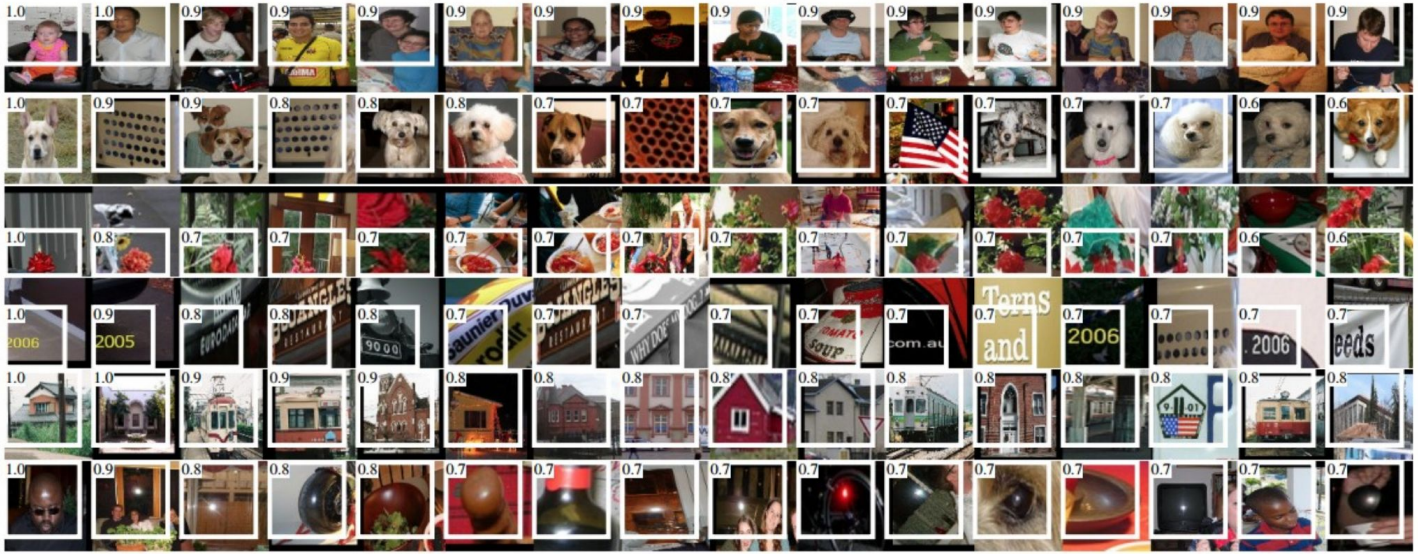


Visualización de los filtros



Visualización de la imagen representante

Nos indica qué imagen es la que consigue la mayor activación para la clase que deseemos. Además, nos indica qué porción de la imagen es la que produce dicha activación (receptive field).



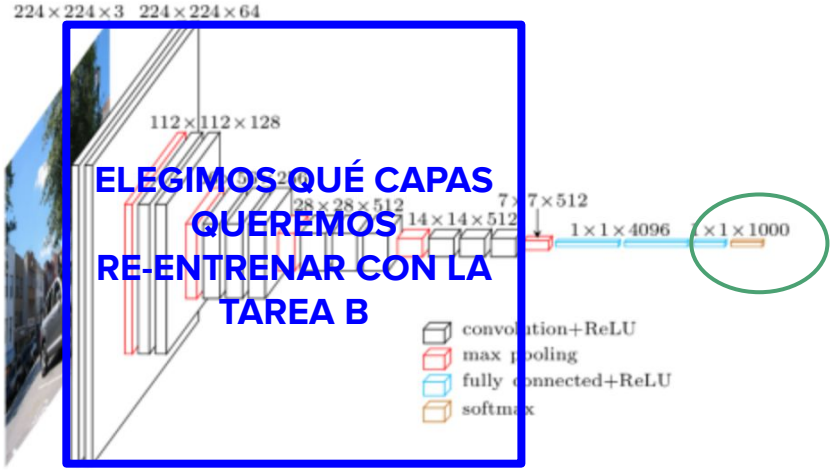
Transfer Learning

Entrenar una red desde 0 es muy costoso en términos de tiempo y de datos necesarios ➜ el Transfer Learning nos permite utilizar una red que ha sido entrenada para una tarea A y aprovechar su conocimiento para solucionar una tarea B

Escogemos las características extraídas por un modelo pre-entrenado en una tarea A para alimentar a un clasificador que entrenamos en otra tarea B.

Fine Tuning

A diferencia del Transfer Learning, con el Fine Tuning sí que re-entrenamos los pesos del modelo pre-entrenado en la tarea A. Podemos elegir si re-entrenamos todos o las últimas N capas. También necesitamos incorporar un nuevo clasificador que pueda trabajar con las clases de nuestro nuevo problema.



Inicializamos nuestro modelo con los pesos de un modelo pre-entrenado en una tarea A, cambiamos el bloque de clasificación para que pueda clasificar el nuevo número de clases y re-entrenamos toda o parte del modelo para la tarea B

¿Cómo elijo qué hago?

Dependiendo del tipo de problema convendrá hacer una cosa u otra. Por ejemplo,, si...

- el nuevo dataset es **pequeño** y **parecido** al original: cuidado al hacer fine-tuning, quizás sea mejor escoger las características de la última capa de la etapa convolucional y usar un SVM o clasificador lineal
- el nuevo dataset es **grande** y **parecido** al original: al tener más datos probablemente no incurramos en over-fitting, así que podemos hacer fine-tuning con más confianza
- el nuevo dataset es **pequeño** y muy **diferente** al original: lo mejor sería usar características de una capa más temprana de la etapa convolucional, ya que ésta se fijará en patrones más generales que las últimas capas, y luego emplear un clasificador lineal
- el nuevo dataset es **grande** y muy **diferente** al original: dale matraca a esa red, entrenala desde el principio! o como se dice en inglés, from scracth! De todas formas, sigue siendo recomendable que inicialices los pesos con los del ImageNet.

Data Augmentation

Permite aumentar el tamaño de nuestro dataset de forma artificial

- Flips horizontales/verticales
- Zooms
- Recortes
- Transformaciones afines
- Transformaciones de color
- Etc.

Por ejemplo:
https://www.wouterbulten.nl/blog/tech/data-augmentation-using-tensorflow-data-dataset/

Data Augmentation