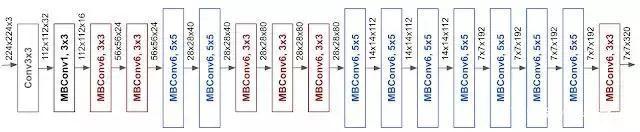
**Punto 5**

Arquitectura EfficientNet

El efecto de la escala del modelo depende en gran medida del modelo de línea de base. Por lo tanto, para mejorar aún más el rendimiento, los investigadores de Google utilizan el marco AutoML MNAS para realizar búsquedas de arquitectura neuronal, desarrollando así un nuevo modelo de línea de base que puede optimizar la precisión y la eficiencia.

El modelo de línea de base utiliza la convolución de cuello de botella invertida móvil (MBConv), similar a MobileNetV2 y MnasNet, pero debido al mayor presupuesto de FLOP, el modelo es más grande. Por lo tanto, los investigadores continuaron escalando el modelo de línea de base para obtener un conjunto de modelos EfficientNets.



El modelo básico EfficientNet-B0 tiene una arquitectura simple y limpia, lo que facilita su expansión y generalización.

Rendimiento EfficientNet

Los investigadores compararon EfficientNets y los modelos CNN existentes en el conjunto de datos ImageNet. El modelo EfficientNet es más preciso y eficiente que el modelo CNN existente, y su cantidad de parámetros y FLOPS se han reducido en un orden de magnitud. Por ejemplo, en el modo de alta precisión, EfficientNet-B7 ha alcanzado la mejor precisión actual del 84,4% entre el 1 y el 97,1% entre los 5 primeros en ImageNet. La velocidad de inferencia de la CPU es 6,1 veces mayor que la de Gpipe, y el tamaño de este último es 8.4 veces de EfficientNet-B7. En comparación con el ResNet-50 ampliamente utilizado, EfficientNet-B4 utiliza FLOPS similares para lograr una precisión de primer nivel que es un 6,3% más alta que ResNet-50 (ResNet-50 76,3%, EfficientNet-B4 82,6%).

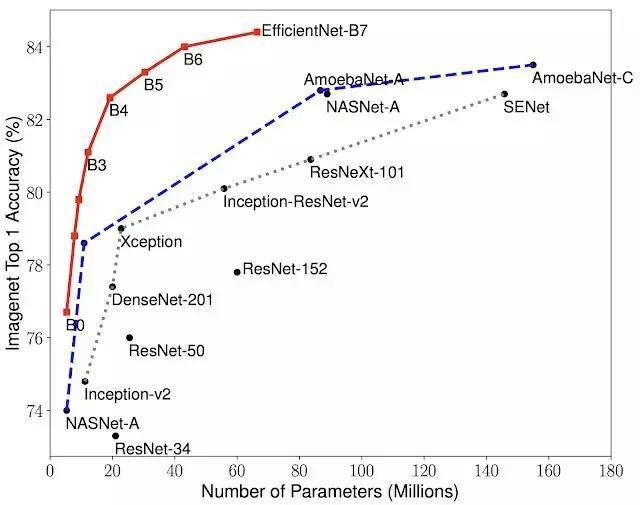


Figura Tamaño del modelo frente a precisión.

EfficientNet-B0 es un modelo de línea de base desarrollado a través de AutoML MNAS, y Efficient-B1 a B7 son redes obtenidas después de extender el modelo de línea de base. EfficientNet es significativamente mejor que otras CNN. Específicamente, EfficientNet-B7 ha logrado nuevos resultados de SOTA: 84,4% de precisión entre 1 y 97,1% entre los 5 primeros, y su tamaño es mucho más pequeño que el modelo óptimo anterior de CNN GPipe (el último tamaño del modelo es EfficientNet- 8.4 veces de B7), la velocidad es 6.1 veces de GPipe. El parámetro de EfficientNet-B1 es mucho más pequeño que el de ResNet-152, pero la velocidad es 5,7 veces mayor que la de este último.

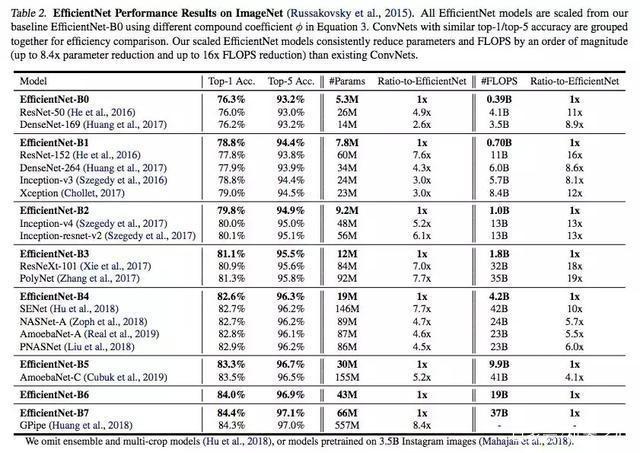
****

Figura El rendimiento de EfficientNet en ImageNet

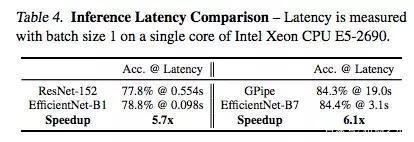


Figura contraste de retardo.

Aunque las EfficientNets funcionan bien en ImageNet, para ser más útiles, deberían tener la capacidad de migrar a otros conjuntos de datos. Los investigadores de Google evaluaron EfficientNets en 8 conjuntos de datos de aprendizaje de migración de uso común. Los resultados muestran que EfficientNets ha alcanzado la tasa de precisión óptima actual en 5 de estos conjuntos de datos, y la cantidad de parámetros se reduce considerablemente, lo que demuestra que EfficientNets tiene buenas capacidades de migración. .

EfficientNets puede mejorar significativamente la eficiencia del modelo, y los investigadores de Google esperan que EfficientNets se pueda utilizar como una nueva base para futuras tareas de visión por computadora. Por lo tanto, los investigadores abrieron el modelo EfficientNet.

El código fuente de EfficientNet y el script de entrenamiento de TPU, consulte: https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

Enlace de referencia: https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

**Punto 6.**

En el siguiente enlace se resuelve un problema de clasificación de imágenes usando una red neuronal convolucional https://poloclub.github.io/cnn-explainer/

a) Describa el ejercicio de clasificación realizado

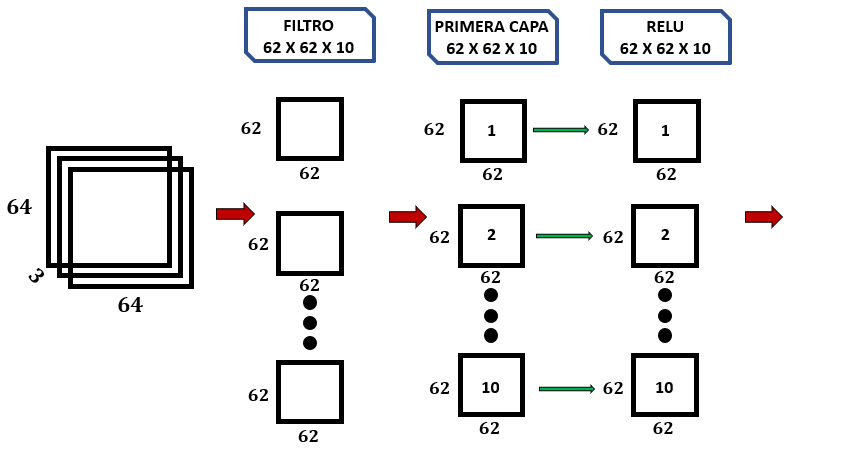
b) Represente gráficamente la red neuronal convolucional usada

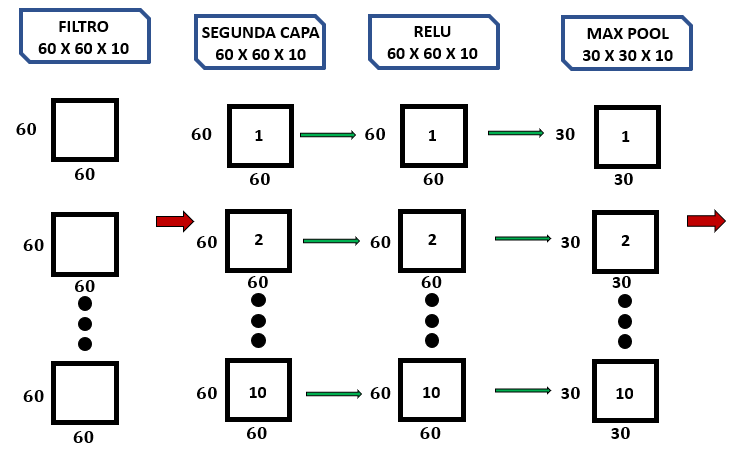
c) Calcule la cantidad de parámetros por cada una de la capas de la red neuronal convolucional utilizada

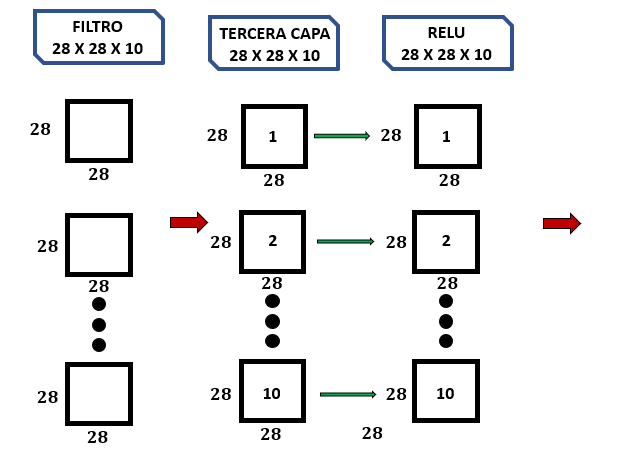
a)

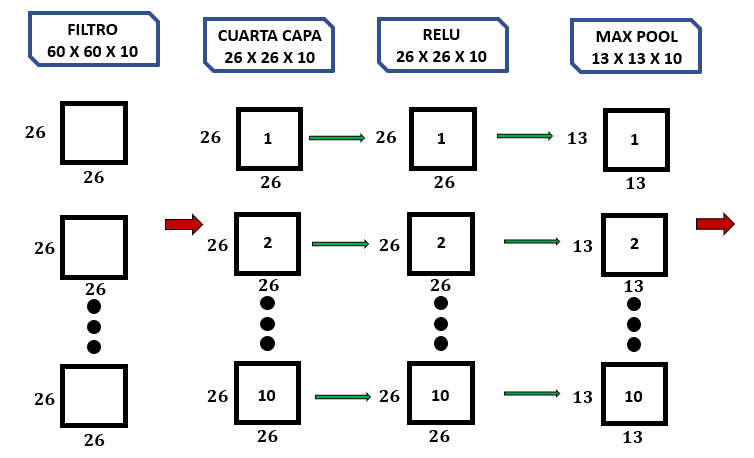
El problema de clasificación de imágenes de CNN Explainer, consta de 10 imágenes en color de tamaño 64x64, las cuales se ubican en 10 clases. Es posible añadir nuevas imágenes a la red, para comprobar su funcionamiento. Las clases que se encuentran en el data set son las siguientes: Lifeboat,. Ladybug ,Pizza,Bell Pepper, School Bus ,Koala. ,Espresso , Red Panda. , Orange. y Sport Car.

b)









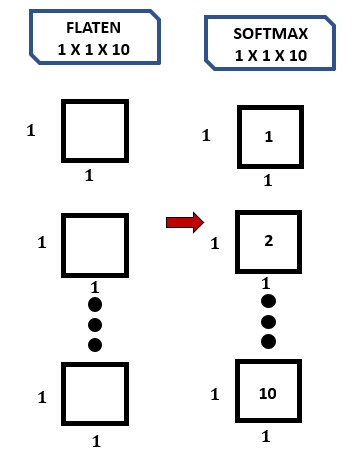


Figura . Arquitectura gráficamente de la CNN Explainer

c) Calcule la cantidad de parámetros por cada una de las capas de la red neuronal convolucional utilizada.

Cálculo de parámetros para la primera Cálculo de parámetros para la primera capa:

Calculo de parámetros para la s Cálculo de parámetros para la segunda capa:

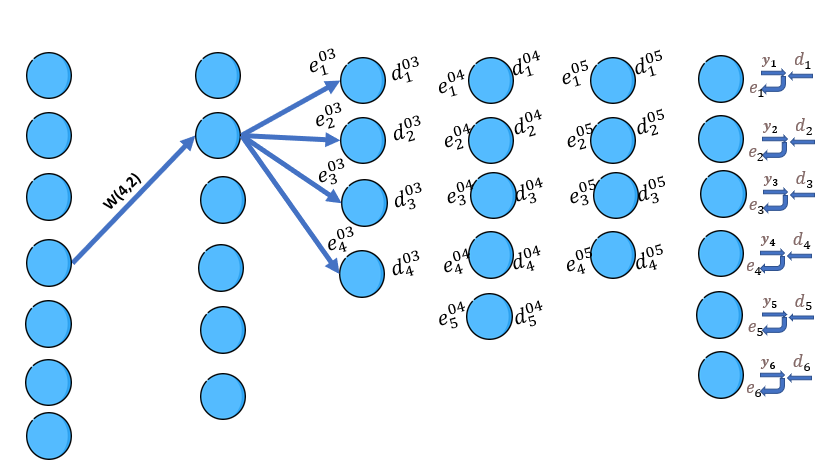
Cálculo de parámetros para la tercera capa:

Cálculo de parámetros para la cuarta capa:

Cálculo de parámetros para la capa de salida:

Cálculo del total de parámetros:

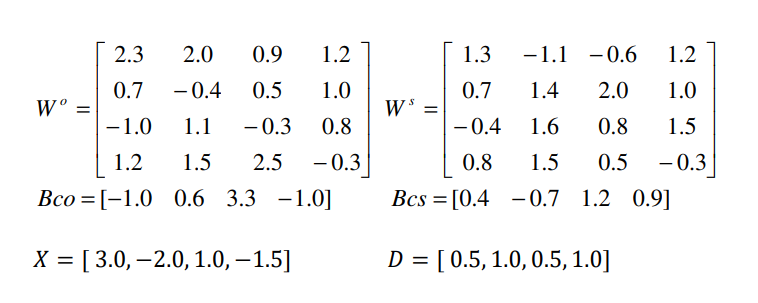
**3.** Se tiene una red neuronal con 6 capas de procesamiento y la siguiente cantidad de neuronas [7 6 4 5 4 6] y las siguientes funciones de activación [lineal, tangente-sigmoidal, simoidal, tangente-sigmoidal, lineal, cúbica], Muestre la expresión para entrenar el peso w(4,2) ubicado entre la primera y segunda capa considerando las funciones de activación definidas.



Hallazgo de error de la capa de salida

Conclusión de expresión :

**4.** Se tiene una red con 4 entradas, 4 neuronas ocultas tangentes sigmoidales y 4 neuronas de salidas sigmoidales. Asuma un valor de razón de aprendizaje igual a 0.60. Encuentre la actualización del valor del bias de la primer neurona de la capa oculta y del peso w(2,4) de la capa de salida.



Referencias

<https://programmerclick.com/article/62541119575/>

<https://arxiv.org/pdf/1905.11946.pdf>