Aprendizaje Automático Profundo (Deep Learning)

Dr. Facundo Quiroga - Dr. Franco Ronchetti

Arquitectura MobileNet

MobileNet (notebook, paper)

- Eficiencia en mente
- Convoluciones “Separables” o “Depthwise”
- Poco uso de memoria y CPU
  - => Mobile
- Relación cantidad feature maps / tamaño similar a VGG
  - 112x112x32 => 56x56x128 => 28x28x256 => 14x14x512 => 7x7x1024
- GlobalAveragePooling para quitar dimensiones espaciales

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

MobileNet (notebook, paper)

- 4 bloques con FMs de 64, 128, 256, 512
- Cada bloque:
  - SeparableConv2D(n,(3,3), stride=(1,1))
  - SeparableConv2D(n,(3,3), stride=(2,2))
- Pero irregulares!
- SeparableConv2D(1024,(3,3)) al final
- AvgPooling
  - Pero le agregan una capa Dense (FC)
- Diseñada para ImageNet
  - Imágenes de 224x224x3
- Disponible en Keras
  - keras.applications.mobilenet.MobileNet()

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Bloque MobileNet (notebook, paper)

- Bloque
  - **n_conv** convoluciones separables depthwise con **stride=1**
  - Convolución separable con stride=2
- OJO! El bloque 1 simplificado

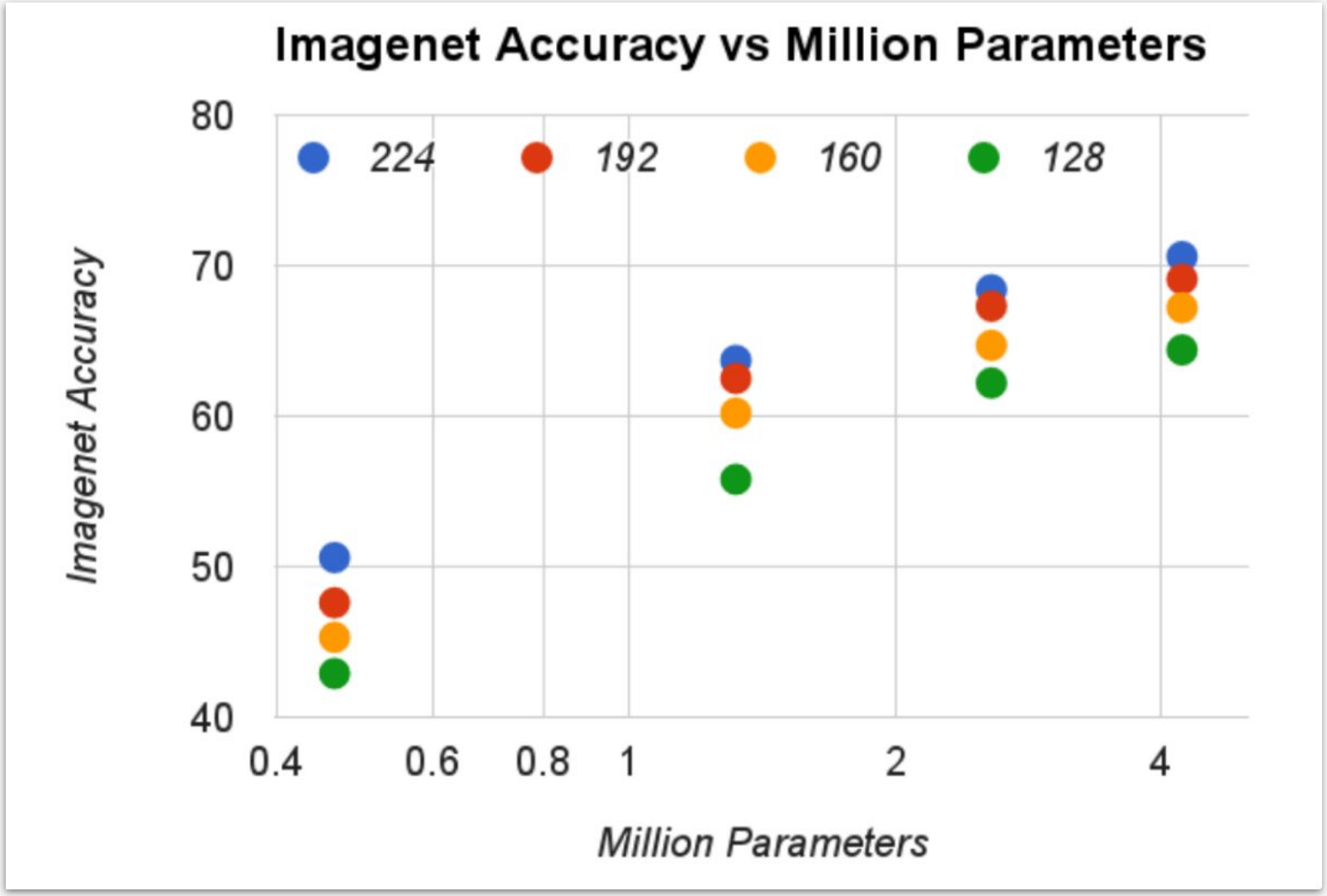
```
def block(n_filters,n_conv):
  layers=[]
  for i in range(n_conv):
    layers.append(SeparableConv2D(n_filters,(3,3),activation="relu",))
  layers.append(SeparableConv2D(n_filters*2,(3,3),
                                activation="relu",strides=(2,2) ))
  return layers
```

Implementación (notebook, paper)

```
def MobileNet(classes,input_shape):
  model = keras.Sequential()
  model.add(InputLayer(input_shape))
  model.add(Conv2D(32,(3,3), activation="relu", padding="same"))


  for n_filters,n_conv in zip([64,128,256,512],[1,1,1,5]):
     layers=block(n_filters,n_conv)
     for layer in layers:
        model.add(layer)
  model.add(SeparableConv2D(1024,(3,3), activation="relu", padding="same"))
  model.add(GlobalAveragePooling2D())
  model.add(Dense(classes))
  model.add(Activation('softmax'))
  return model
```

Accuracy vs resolución vs parámetros (ImageNet)



- **keras.applications.mobilenet.MobileNet()** tiene un hiperparámetro llamado **alpha**
- alpha = 1: red común
- alpha > 1: más filtros
- alpha < 1: menos filtros
- Ejemplo:
  - alpha = 0.5
    - Mitad de filtros
    - (mitad de param)

Accuracy vs N° de operaciones (ImageNet)