Aprendizaje Automático Profundo (Deep Learning)

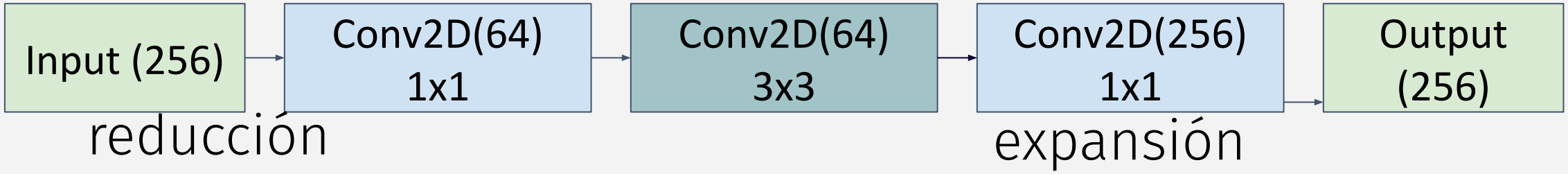
Dr. Facundo Quiroga - Dr. Franco Ronchetti

Arquitectura Residual Network (ResNets)

Redes residuales

Bloques Bottleneck

- Objetivo: menor coste computacional
  - 1x1x256 + 3x3x64 + 1x1x+256 = 1088 parámetros vs 3x3x256=2304



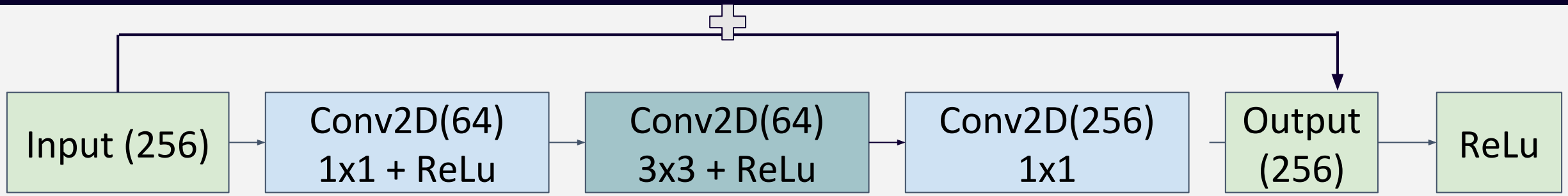

```
def bottleneck(x,n_reduction,n_expansion):
    x = Conv2D(n_reduction,(1,1),padding="same",activation="relu")(x)
    x = Conv2D(n_reduction,(3,3),padding="same",activation="relu")(x)
    x = Conv2D(n_expansion,(1,1),padding="same",activation="relu")(x)
    return x
```

Bloques Residuales

- Bloques residuales
  - Aprenden modificación (vs transformación) de la entrada
  - Al comenzar entrenamiento, la capa es la función identidad
    - La salida es igual a la entrada (mismas dimensiones)
  - Puedo poner muchas más capas
    - En el peor caso, no hacen ninguna modificación.


```
def residual(x):
  x_mod = Capa(...)(x)
  return x + xmod
```

Bloques Conv Bottleneck + Residuales




```
def residual_bottleneck(x,n_reduction,n_expansion):
  x_mod=x
  x_mod = Conv2D(n_reduction,(1,1),activation="relu")(x_mod)
  x_mod = Conv2D(n_reduction,(3,3),activation="relu")(x_mod)
  x_mod = Conv2D(n_expansion,(1,1),activation=None)(x_mod)
  new_x = Add()([x, x_mod])
  new_x = Activation("relu")(new_x) # mismas dimensiones
  return new_x
```

Modelo ResNet (Red Residual) (paper,notebook)

- Primeras en entrenar redes con hasta 152 capas

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Modelo ResNet (Red Residual) (notebook)

```
def ResNet50(input_shape,classes):
  input = Input(shape=input_shape)
  x = Conv2D(64,(7,7),strides=(2,2),padding="same",activation="relu")(input)
  reductions = [64,128,256,512]
  blocks     = [3,4,6,3]
  for i,(block_n,reduction) in enumerate(zip(blocks,reductions)):
    expansion=reduction*2
    x = Conv2D(expansion,(3,3),strides=(2,2),padding="same",activation="relu")(x)
    for j in range(block_n):
      x = residual_bottleneck(x,reduction,expansion)
  x = GlobalAveragePooling2D(name="gap")(x)
  x = Dense(classes,activation="softmax")(x)
  model = Model(inputs=[input],outputs=[x])
  return model
```