

Aprendizaje Profundo

Práctica 2: Residual Neural Networks

11 de noviembre de 2024

Se ha entrenado una Residual Neural Network sobre el dataset CIFAR-100, cuya arquitectura está definida en la Figura 1. Dicha arquitectura consta de las siguientes especificaciones:

Model: "sequential_3"

Layer (type)	Output Shape	Param #
conv2d_22 (Conv2D)	(None, 32, 32, 16)	432
residual_block_9 (ResidualBlock)	(None, 32, 32, 64)	47,424
residual_block_10 (ResidualBlock)	(None, 32, 32, 64)	74,240
residual_block_11 (ResidualBlock)	(None, 32, 32, 64)	74,240
residual_block_12 (ResidualBlock)	(None, 16, 16, 128)	230,144
residual_block_13 (ResidualBlock)	(None, 16, 16, 128)	295,936
residual_block_14 (ResidualBlock)	(None, 16, 16, 128)	295,936
residual_block_15 (ResidualBlock)	(None, 8, 8, 256)	919,040
residual_block_16 (ResidualBlock)	(None, 8, 8, 256)	1,181,696
residual_block_17 (ResidualBlock)	(None, 8, 8, 256)	1,181,696
batch_normalization_37 (BatchNormalization)	(None, 8, 8, 256)	1,024
activation_19 (Activation)	(None, 8, 8, 256)	0
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 256)	0
dense_1 (Dense)	(None, 100)	25,700

Total params: 4,327,508 (16.51 MB)

Trainable params: 4,322,100 (16.49 MB)

Non-trainable params: 5,408 (21.12 KB)

Figura 1: Arquitectura general de la red.



- Los datos de entrada tienen que estar escalados entre los valores $[-1..1]$.
- Comienza con una Convolución 3x3.
- Ninguna de las convoluciones de esta red **tiene el vector de bias**.
- Se ha utilizado SiLU como función de activación en todas las capas intermedias.
- La capa ResidualBlock tiene un comportamiento variable en función de número de canales en la entrada y en la salida, tal y como se puede ver en la Figura 2:
 - Si el número de canales a la entrada es igual al número de canales a la salida, se utiliza la conexión de salto clásica.
 - En cambio, si difieren, se aplica una convolución 1x1 sobre la conexión residual.

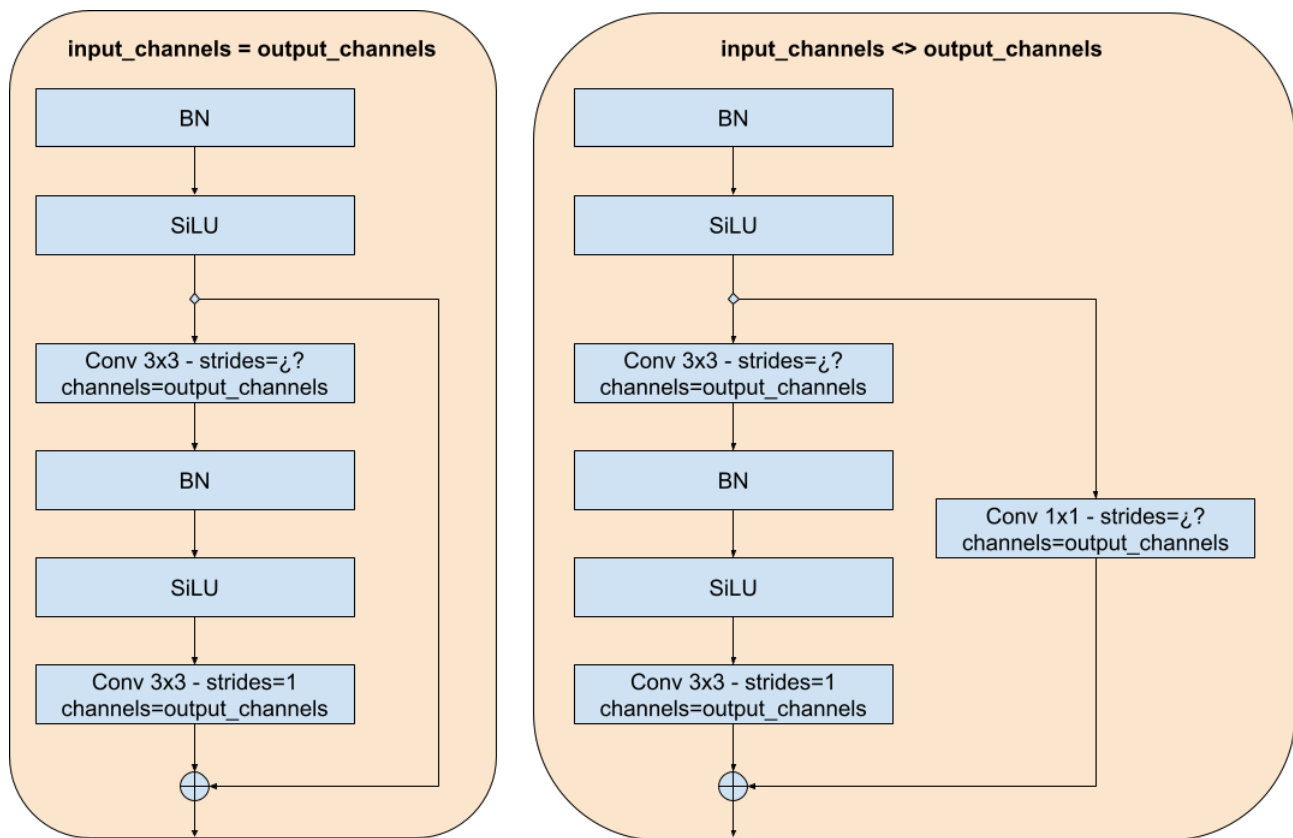


Figura 2: Arquitectura de la capa ResidualBlock.

```
class ResidualBlock(Model):

    def __init__(self, input_channels, output_channels, strides=(1, 1)):
        ...

    def call(self, x):
        ...
```

Figura 3: Plantilla para la clase ResidualBlock.

Trabajo a realizar

- (2 puntos) Define la capa ResidualBlock (ver Figura 2), usando como base la plantilla proporcionada en la Figura 3.
 - Ten en cuenta que el número de convoluciones depende de los valores de input_channels y output_channels.
 - Esta red no tiene capas de Pooling. La reducción del tamaño se realiza con el parámetro strides de las convoluciones, pero se modifica únicamente en 1 (o 2) de las convoluciones del modelo.
- (2 puntos) Define la red ResidualNetwork (ver Figura 1). Para comprobar su correcto funcionamiento haz lo siguiente:
 - Descarga los pesos del modelo preentrenado (los podrás encontrar en el canal de Teams de la asignatura).
 - Carga los pesos en tu modelo, haciendo uso de la función proporcionada en la Figura 4.
 - Comprueba que la precisión del modelo en CIFAR-100 es superior al 69 %.

3. (4 puntos) Entrena, mediante la técnica de fine-tuning, sobre el dataset CIFAR-10, manteniendo fijos los pesos de la red preentrenada proporcionada en el canal de Teams de la asignatura. Analiza el resultado en el conjunto de test. Se penalizarán los siguientes puntos:
- (-4 puntos) No se ha entrenado la red correctamente, y no se proporciona el resultado obtenido en el conjunto de test.
 - (-1 punto) No se ha utilizado la red original completa, a excepción de la última capa.
 - (-1 punto) No se optimizado el entrenamiento, reduciendo al máximo el consumo de memoria.
 - (-1 punto) No se ha hecho uso de técnicas de `DataAugmentation` sobre las imágenes de entrenamiento.
 - (-2 puntos) No se ha entrenado la red sin hacer uso de la función `fit`, entrenando el modelo con un bucle de entrenamiento desde cero.
4. (4 puntos) Entrena, mediante la técnica de fine-tuning, sobre el dataset CIFAR-10, sin mantener fijos los pesos de la red preentrenada proporcionada en el canal de Teams de la asignatura. Analiza el resultado en el conjunto de test. Se penalizarán los siguientes puntos:
- (-4 puntos) No se ha entrenado la red correctamente, y no se proporciona el resultado obtenido en el conjunto de test.
 - (-1 punto) No se ha utilizado la red original completa, a excepción de la última capa.
 - (-1 punto) No se han mantenido congeladas las capas de `BatchNormalization`.
 - (-1 punto) No se ha hecho uso de un learning rate muy bajo, con el objetivo de no perder las capacidades de generalización de los pesos preentrenados.
 - (-2 puntos) No se ha entrenado la red sin hacer uso de la función `fit`, entrenando el modelo con un bucle de entrenamiento desde cero.

```
import pickle

def load_weights(model, weight_file):
    with open(weight_file, 'rb') as f:
        weights = pickle.load(f)

    all_vars = model.trainable_weights + model.non_trainable_weights
    weight_list = [(x, weights[x]) for x in sorted(weights.keys())]
    weights = {}
    for i, var in enumerate(all_vars):
        aux = var.path.split('/')[2:]
        classname = '_'.join(aux[0].split('_')[:-1])
        name = aux[1]
        assigned = False
        for j, (key, value) in enumerate(weight_list):
            if classname in key and name in key:
                try:
                    all_vars[i].assign(value)
                except:
                    continue
                print('assigning', key, 'to', var.path)
                del weight_list[j]
                assigned = True
                break
    if not assigned:
        raise Exception(var.path + ' cannot be loaded')
```

Figura 4: Plantilla para cargar los pesos de la red preentrenada.



- Se entregarán en el mismo *assignment* de Classroom que la práctica 1 (<https://classroom.github.com/a/WdUBDmBc>) como uno o varios cuadernos nuevos. Puedes organizarlos en directorios si lo consideras necesario.
- Se deberá crear un tag llamado **P2**. El contenido de este tag será lo que se evalúe.
- La fecha de entrega límite es el día **29 de noviembre de 2024** a las 23:50 horas.
- Se valorarán las explicaciones de los resultados:
 - El proceso está suficientemente detallado y las decisiones adoptadas están justificadas.
 - Los resultados se comentan e interpretan correctamente.