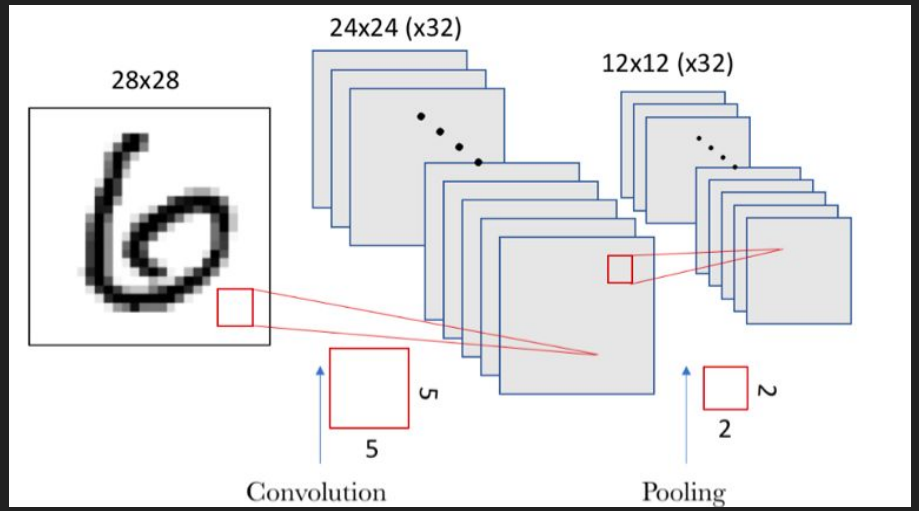
Redes neuronales convolucionales con Keras

Clasificación de imágenes de gestualización numérica decimal con una mano

Keras

I. Introducción

Redes convolucionales



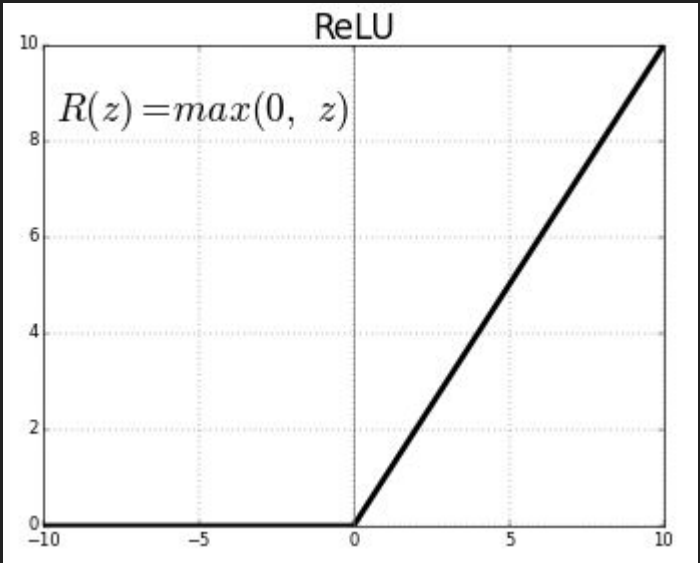
Las redes neuronales convolucionales son las más adecuadas para el procesamiento de imágenes.

Estas redes son capaces de analizar una imagen obteniendo patrones y con estos aprender conceptos abstractos.

II. Preliminares

Métodos empleados

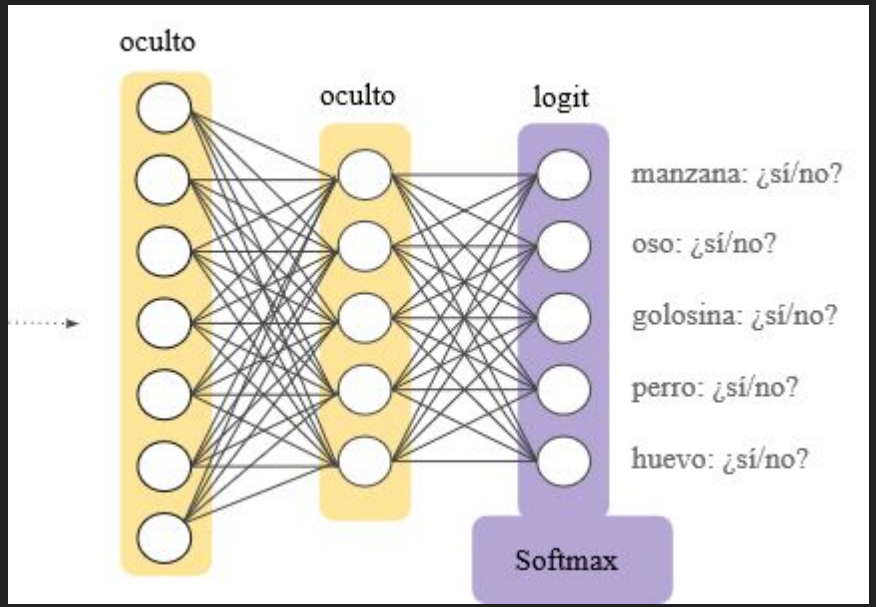
- Función de activación ReLu



Si el valor de entrada es negativo, el valor de salida ser 0.
Si no es así, el valor de entrada será el valor de salida.

Métodos empleados

- Clasificador probabilístico Softmax

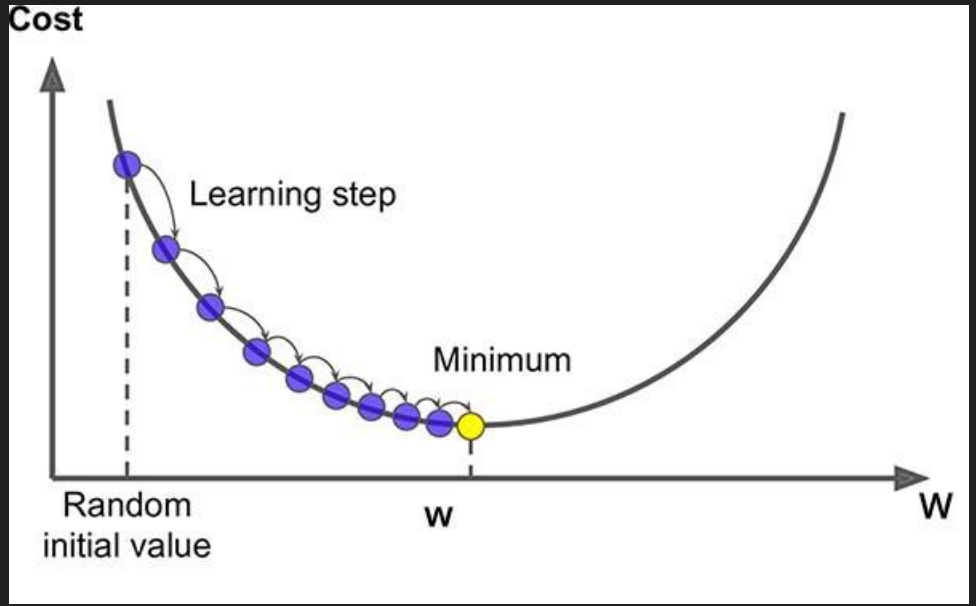


$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

Devuelve un vector con la probabilidad de que la imagen pertenezca a cada clase.

Métodos empleados

- Optimizador de descenso del gradiente estocástico



El gradiente es un cálculo que nos permite saber cómo ajustar los parámetros de la red de tal forma que se minimice su desviación a la salida.

Trabajo relacionado

- “Very Deep Convolutional Networks for Large-Scale Image Recognition” (VGG16)

- “ImageNet Classification with Deep Convolutional Neural Networks” (AlexNet)

III. Metodología

Metodología

Con el objetivo de hacer un **análisis comparativo entre distintas redes convolucionales** y su rendimiento, se proponen **2 arquitecturas distintas:**

CarlosNet

IvanNet

Para evaluar el rendimiento de estas redes se las ha entrenado desde el principio 2 veces:

1. Con el **Dataset base**
2. Con un **Dataset aumentado**

Las métricas que se han utilizado para comparar los modelos:

- **Curvas de aprendizaje**
- **Valores de precisión** para los subconjuntos train, val y test
- **Valores de la función de pérdida** para los subconjuntos de train, val y test.

Dataset original

Este dataset contiene un total de **21600 imágenes** que a su vez están divididas en los **3 subconjuntos:**

- Entrenamiento: **4/6 del conjunto**, unas 14400 imágenes.
- Validación: **1/6 del conjunto**, unas 3600 imágenes.
- Prueba: **1/6 del conjunto**, unas 3600 imágenes.

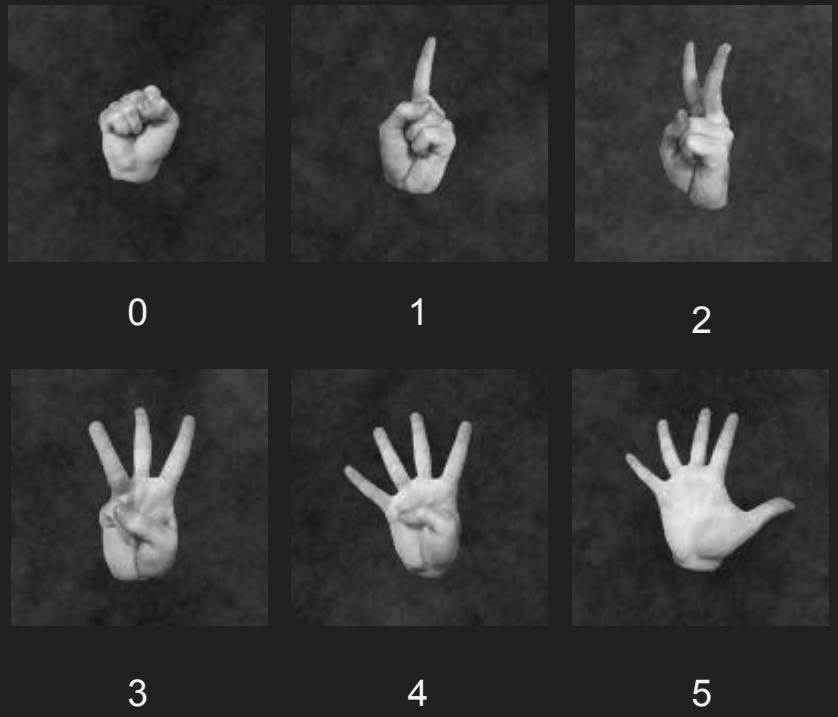
Dataset original

Se compone de imágenes de **tamaño 128x128** y en **blanco y negro**.

Las imágenes se dividen en **6 clases**, dependiendo del valor que representan los dedos.

El conjunto es **bastante homogéneo**:

- Mano centrada
- Contraste con el fondo oscuro
- Orientación siempre vertical



Dataset aumentado

Partiendo del dataset base, éste se transforma para obtener una **diversidad de éste mucho mayor**.

- Transformación espejo vertical / horizontal
- Rotaciones de hasta 90º
- Desplazamiento hasta de un 20% del tamaño total
- Zoom entre [+20%,-20%]

Se introduce una **mayor heterogeneidad** pero **sin alterar excesivamente el dataset original**.

Dataset aumentado

Parámetros utilizados

- horizontal_flip=True
- vertical_flip=True
- rotation_range=90
- width_shift_range=0.2
- height_shift_range=0.2
- zoom_range=0.2

Arquitectura de las redes

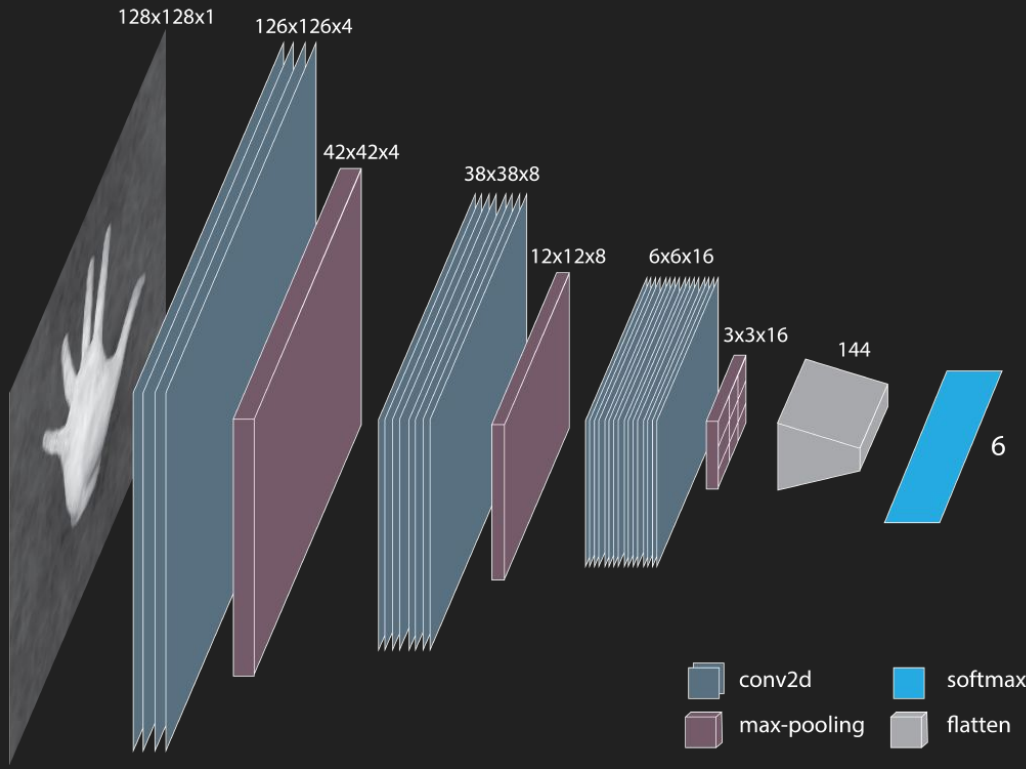
Nuestra **heurística** a la hora de definir los hiper parámetros.

1. El **tamaño** de los filtros **pequeño al principio** para conseguir la mayor cantidad de información local posible, para después **aumentar gradualmente el tamaño** de éstos.
2. La **profundidad** de las convoluciones debe ser **baja al principio** para detectar **detalles de bajo nivel**, y luego aumentar con el fin de extraer características, formas y **patrones más complejos** de la imagen.
3. Ventanas impares.

Arquitectura de las redes: CarlosNet

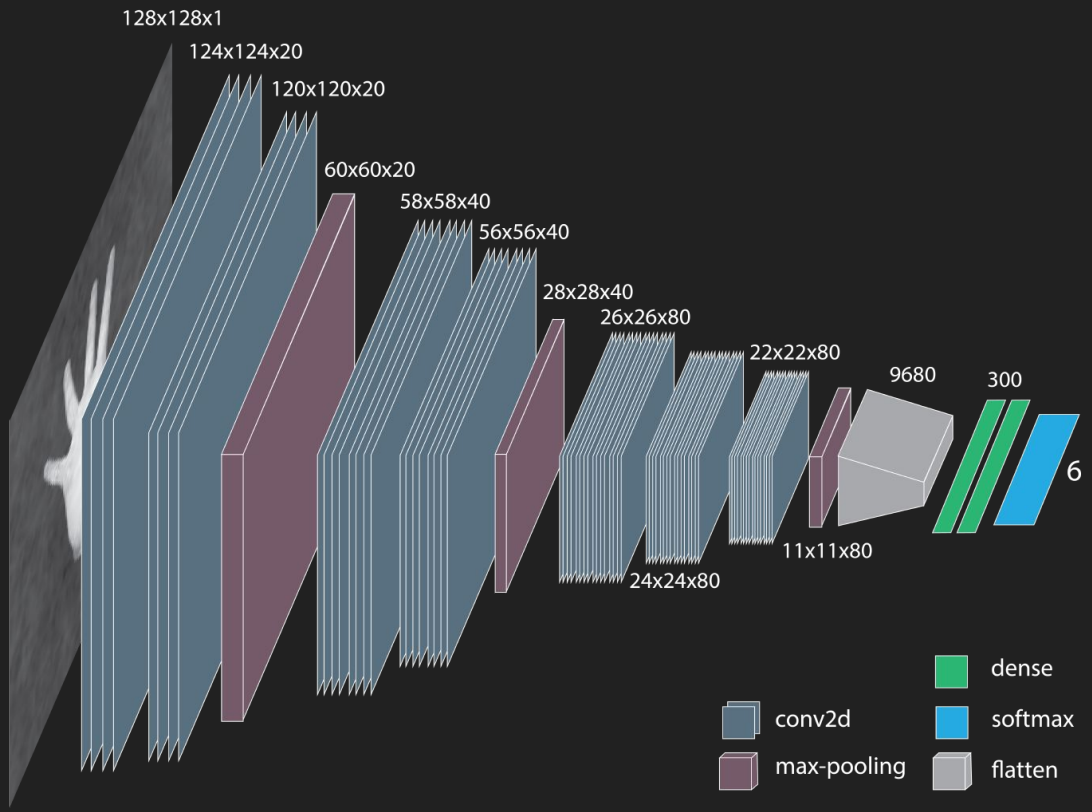
1. Entrada: Tensor de 128x128x1.
2. Convolución1: Ventana de 3x3 con 4 filtros.
3. Max-Pooling1: Ventana de 3x3.
4. Convolución2: Ventana de 5x5 con 8 filtros.
5. Max-Pooling2: Ventana de 3x3.
6. Convolución3: Ventana de 7x7 con 16 filtros.
7. Max-Pooling3: Ventana de 2x2.
8. Flatten: 9680 parámetros
9. Softmax: 144 entradas.
10. Salida: Vector de 6 valores [0.,1.]

8000 parámetros de entrenamiento



Arquitectura de las redes: IvanNet

1. Entrada: Tensor de 128x128x1.
2. Convolución1: Ventana de 5x5 con 20 filtros.
3. Convolución2: Ventana de 5x5 con 20 filtros.
4. Max-Pooling1: Ventana de 2x2.
5. Convolución3: Ventana de 3x3 con 40 filtros.
6. Convolución4: Ventana de 3x3 con 40 filtros.
7. Max-Pooling2: Ventana de 2x2.
8. Convolución5: Ventana de 3x3 con 80 filtros.
9. Convolución6: Ventana de 3x3 con 80 filtros.
10. Convolución7: Ventana de 3x3 con 80 filtros.
11. Max-Pooling3: Ventana de 2x2.
12. Flatten: 9680 parámetros
13. Densa1: 9680 entradas
14. Densa2: 600 entradas
15. Softmax: 300 entradas
16. Salida: Vector de 6 valores [0.,1.]



6 167 166 parámetros entrenables.

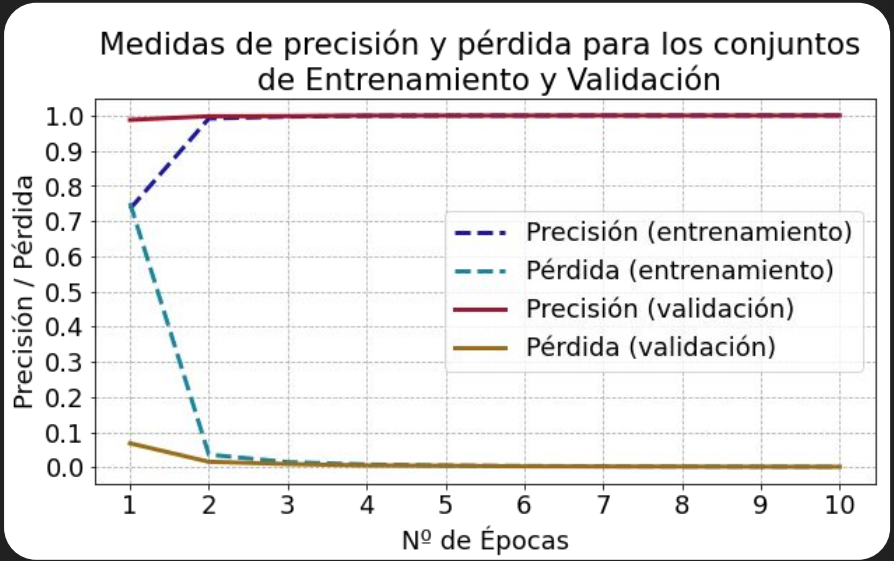
IV. Resultados

Entrenamiento con el dataset original

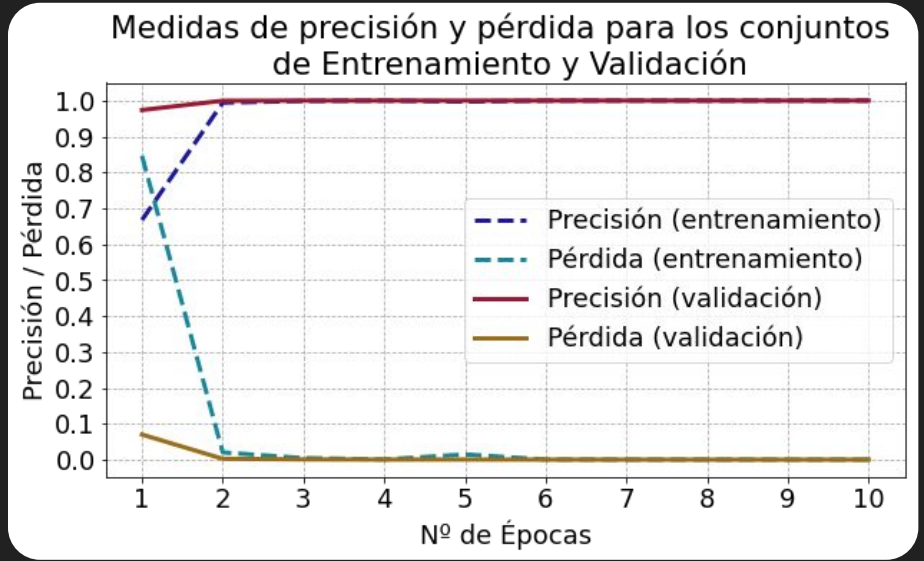
En la primera fase de experimentación, se entrenaron los modelos con el dataset tal y como se proporciona.

Ambas redes se entrenaron en esta fase con **10 épocas**.

Entrenamiento con el dataset original



CarlosNet



IvanNet

Resultados del entrenamiento con el dataset original

	Precisión (train)	Precisión (val)	Precisión (test)
CarlosNet	1.0	1.0	0.9994
IvanNet	1.0	1.0	1.0

	Pérdida (train)	Pérdida (val)	Pérdida (test)
CarlosNet	0.0014	0.0012	0.0031
IvanNet	2.8292e-5	2.076e-5	5.825e-5

Entrenamiento con el dataset aumentado

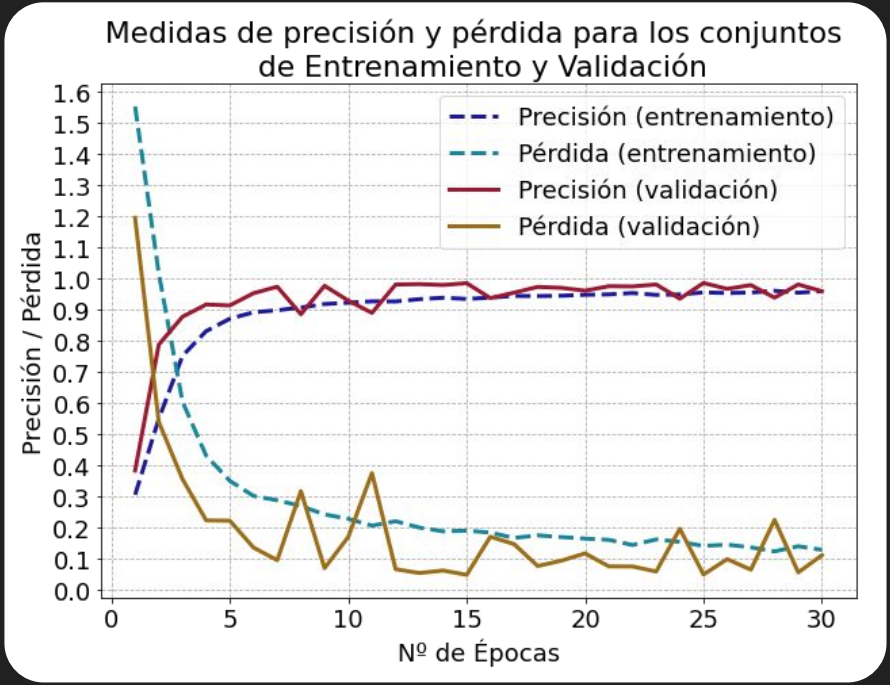
En la segunda fase de experimentación, se ha entrenado los modelos con el **dataset aumentado**.

Ambas redes se entrenaron en esta fase con **30 épocas**.

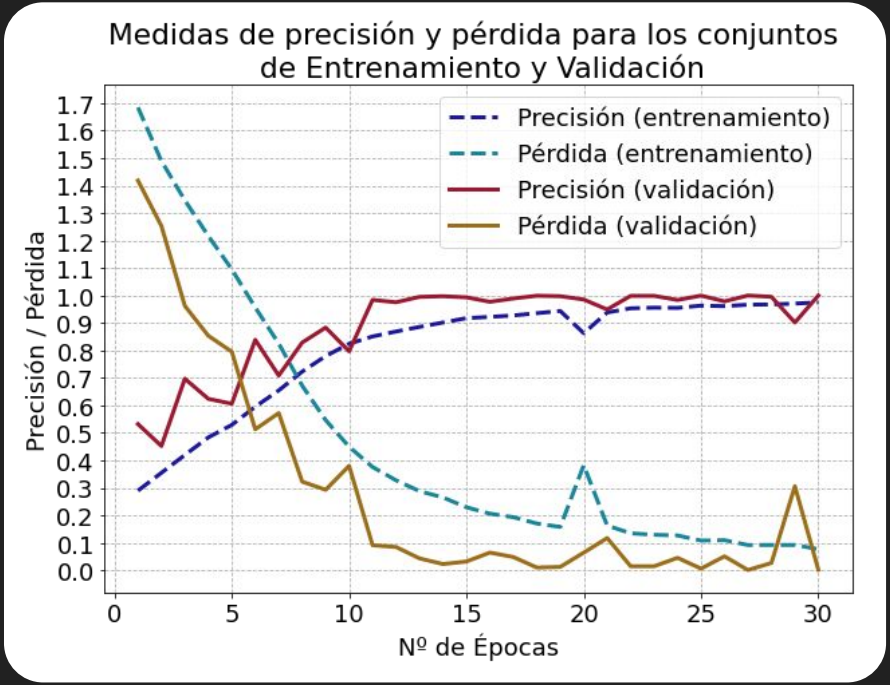
En las gráficas de ambos modelos se evidencian un par de fenómenos interesantes que antes no se producían:

1. La existencia de **picos u oscilaciones** de la función de pérdida respecto al conjunto de validación.
2. La función de pérdida del conjunto de validación **se mantiene casi siempre por debajo de la función de pérdida** el conjunto de entrenamiento.

Entrenamiento con el dataset aumentado



CarlosNet



IvanNet

Resultados del entrenamiento con el dataset aumentado

	Precisión (train_aug)	Precisión (val)	Precisión (test)
CarlosNet	0.9587	0.9586	0.9767
IvanNet	0.9823	0.9942	0.9955

	Pérdida (train_aug)	Pérdida (val)	Pérdida (test)
CarlosNet	0.1272	0.1099	0.0647
IvanNet	0.0455	0.0017	0.00955

Comparación de la capacidad de generalización

	Precisión (train_aug)	Precisión (val_aug)	Precisión (test_aug)
CarlosNet	0.1972	0.1978	0.1917
CarlosNet (aug)	0.9587	0.9586	0.9767

	Precisión (train_aug)	Precisión (val_aug)	Precisión (test_aug)
IvanNet	0.1981	0.1975	0.1997
IvanNet (aug)	0.9823	0.9942	0.9955

V. Conclusiones

- Hacer data augmentation ha sido muy útil para aumentar la generalización de nuestro modelo en ambas arquitecturas desarrolladas.Los modelos se siguen adaptando muy bien a la clasificación del dataset original.

- Con ayuda de las gráficas observamos que al hacer data augmentation y validar con datos del dataset original la precisión es generalmente mayor durante la validación que durante el entrenamiento, exceptuando algunos picos, pero este hecho no repercute en el rendimiento general del modelo.