



ESCUELA DE  
INFORMÁTICA Y  
TELECOMUNICACIONES

# ETIQUETADO DE IMAGENES PARA GOBIERNO DE CHILE

Integrantes:

CRISTOBAL CABEZAS

JORGE LOPEZ S.

Profesor: MARCO JAPKE



# Business Understanding

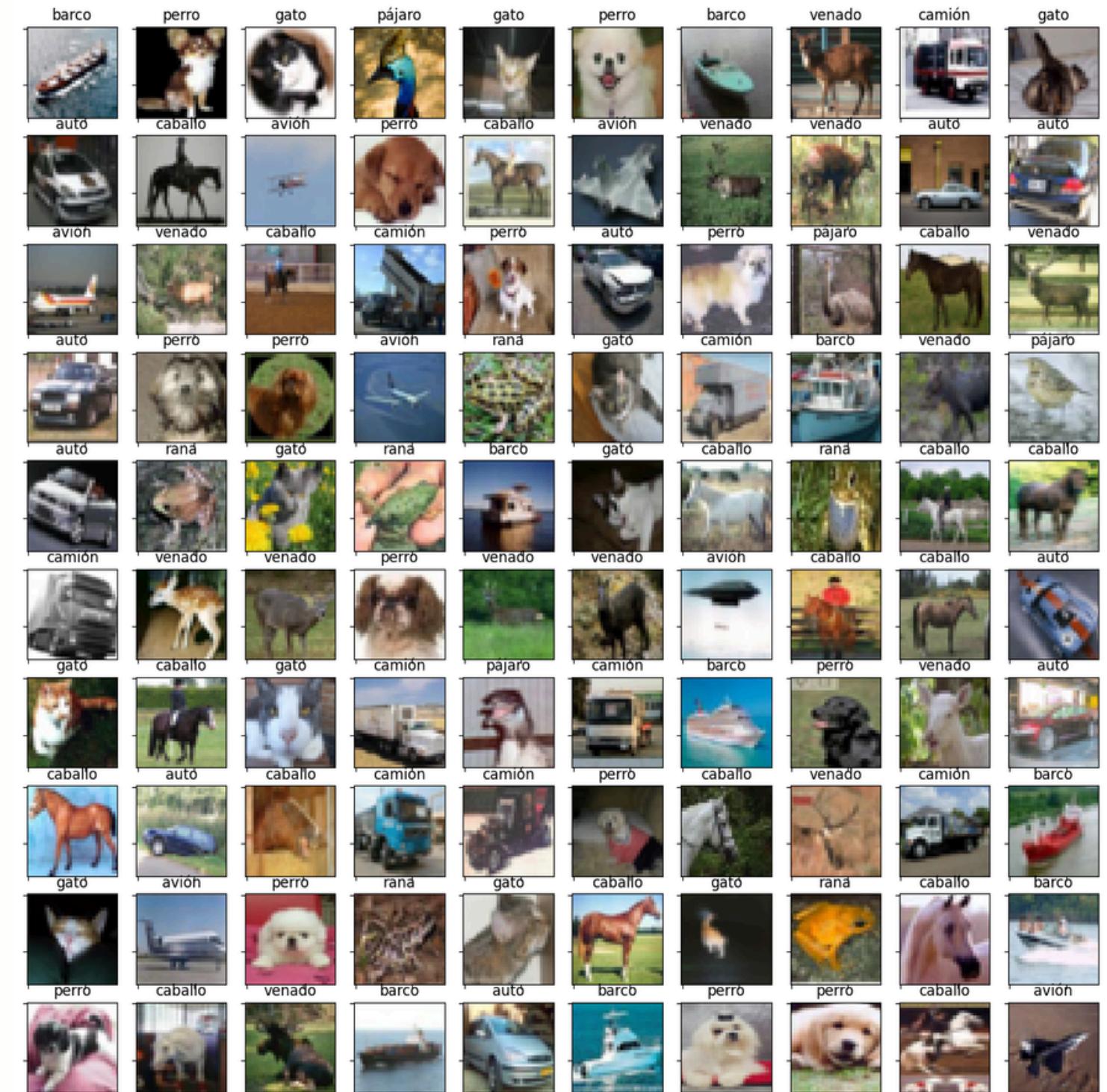
Etiquetado de imágenes CIFAR-10

El Gobierno de Chile busca implementar inteligencia artificial en su página web mediante el etiquetado de imágenes de diversos dominios. Se destaca la importancia de la Inteligencia Artificial en la actualidad para resolver problemas de clasificación, detección de objetos, estados de ánimo y reconocimiento facial, especialmente en un contexto de gran cantidad de datos.

.....

# Data Understanding

- 60000 imágenes en color
- 32x32 píxeles
- Divididas en 10 clases (etiquetas),
- 6000 imágenes por clase para entrenamiento y  
10000 para pruebas



# Data Preparation:

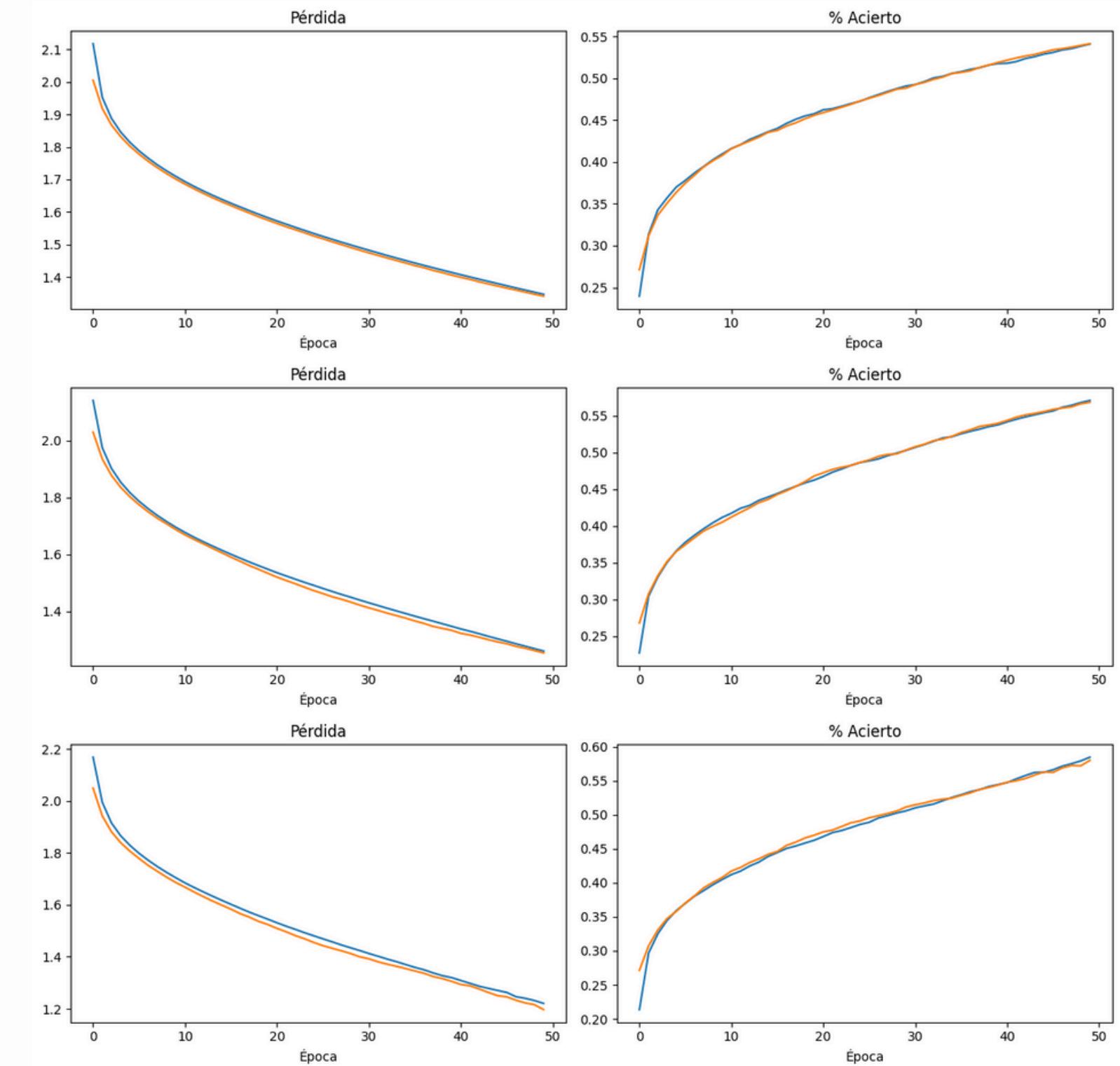
- Transformación de datos para entrenamiento
  - reshape
  - to\_categorical
- Se definen 3 capas:
  - Capa 1 --> 512 neuronas
  - Capa 2 --> 256 neuronas
  - Capa 3 --> 256 neuronas



# Modeling

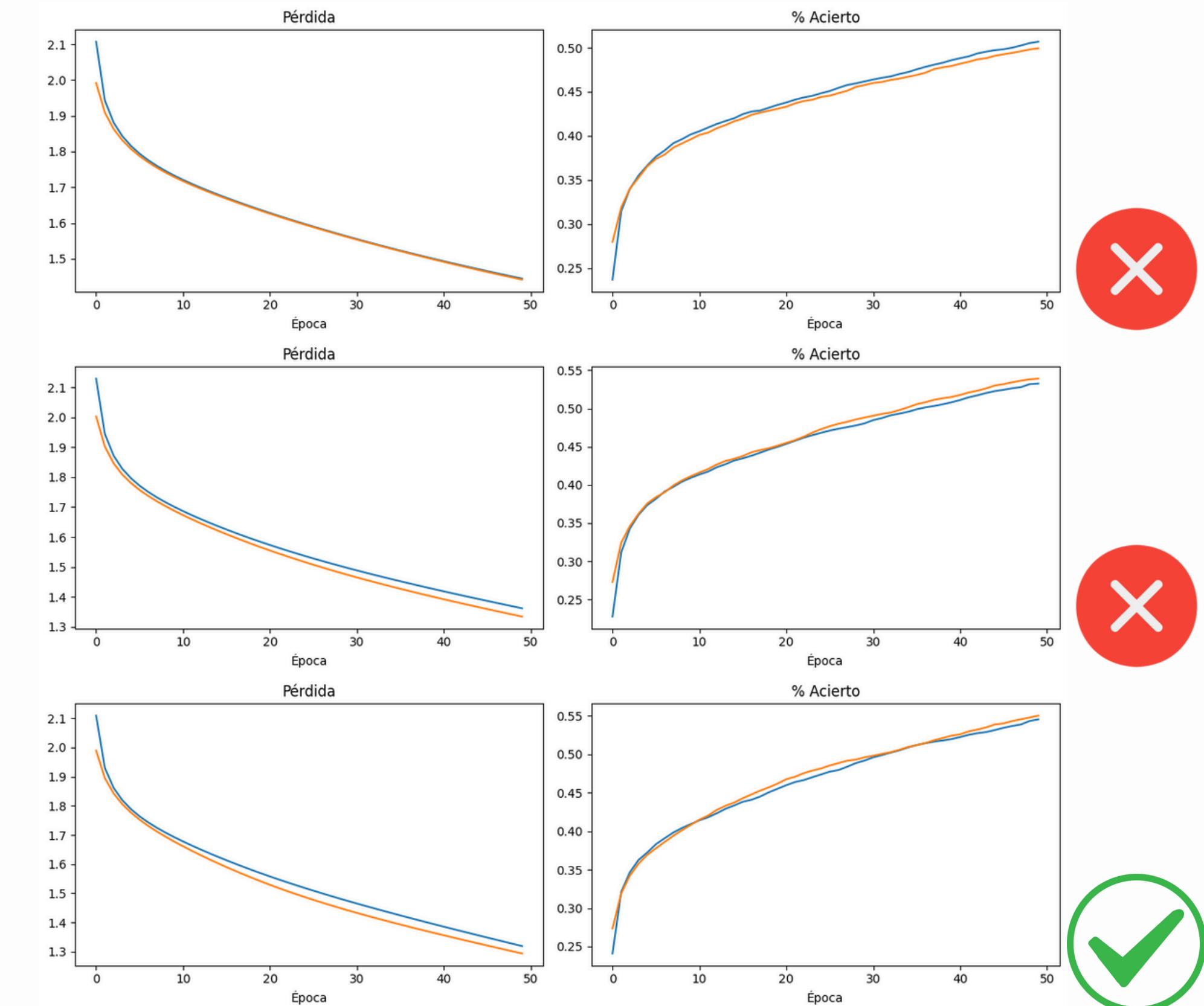
# SGD + ReLu

Relu					
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	0,5448	1,3521	0,5414	1,3416	82
2	0,5731	1,2609	0,5685	1,2531	60
3	<b>0,59</b>	<b>1,2237</b>	<b>0,5802</b>	<b>1,1967</b>	<b>60</b>



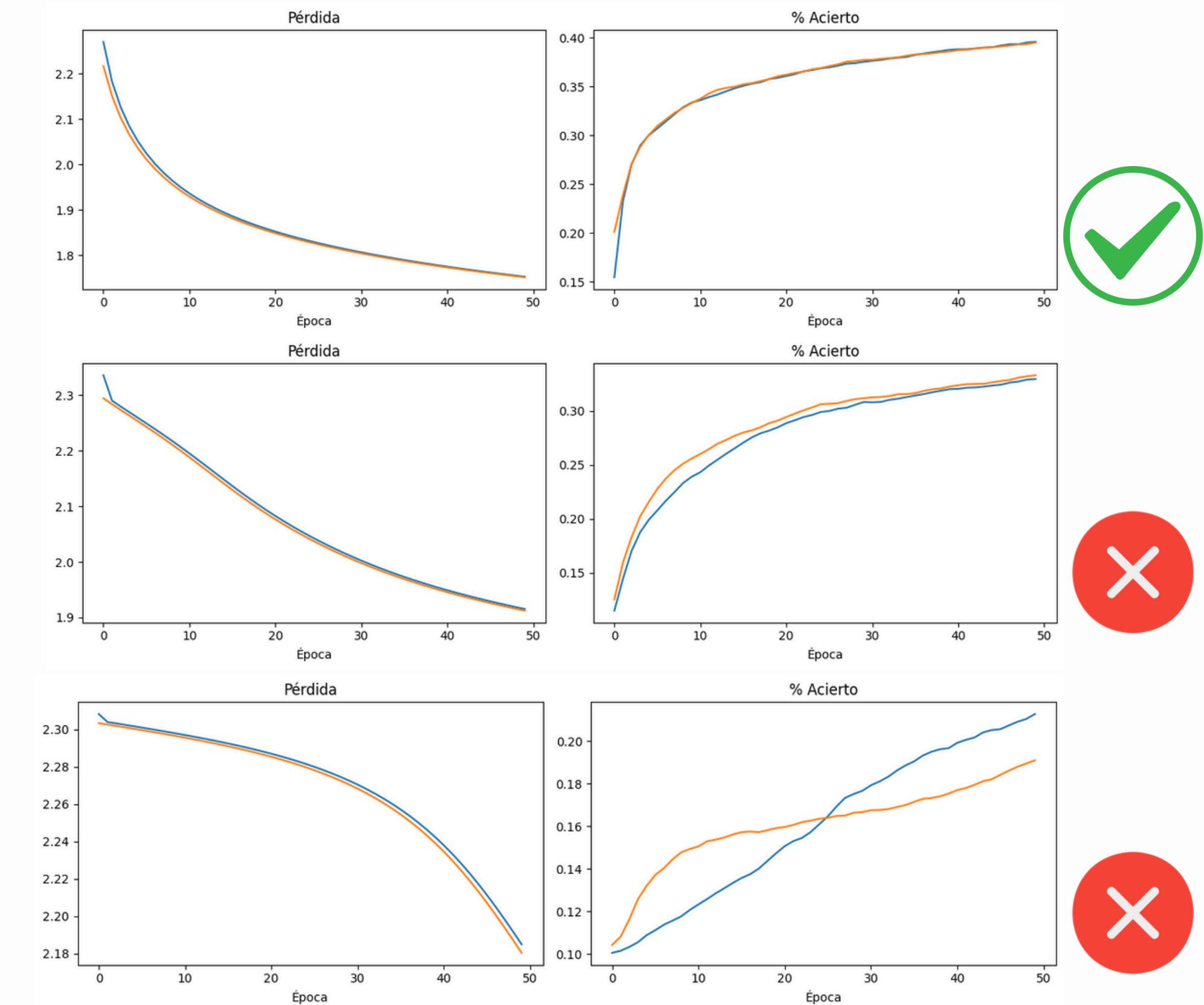
# SGD + Tanh

Tanh					
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	0,5083	1,4454	0,4993	1,4415	62
2	0,5334	1,3618	0,5391	1,3339	65
3	<b>0,5454</b>	<b>1,3208</b>	<b>0,5503</b>	<b>1,2933</b>	<b>68</b>



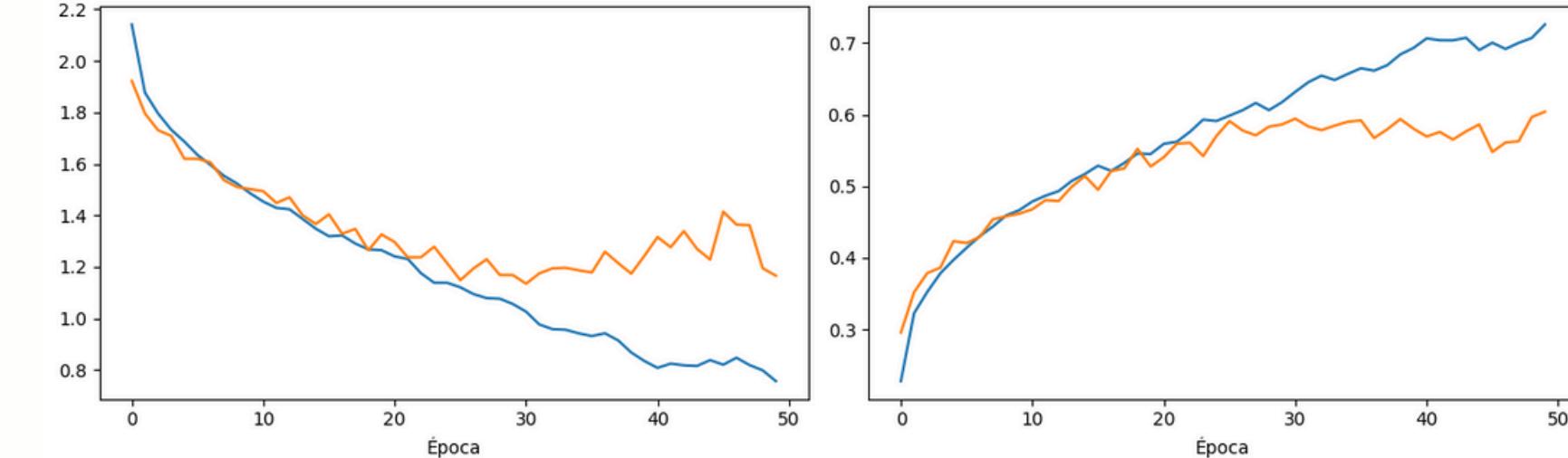
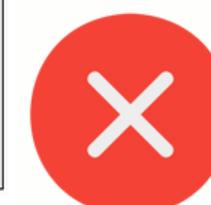
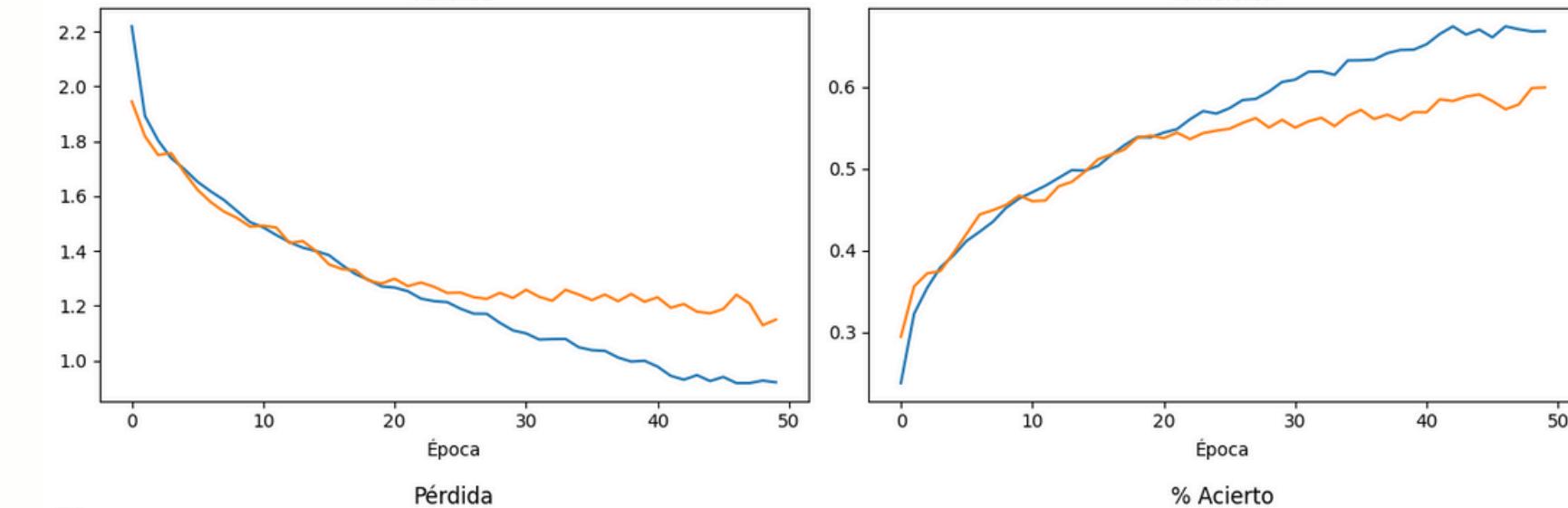
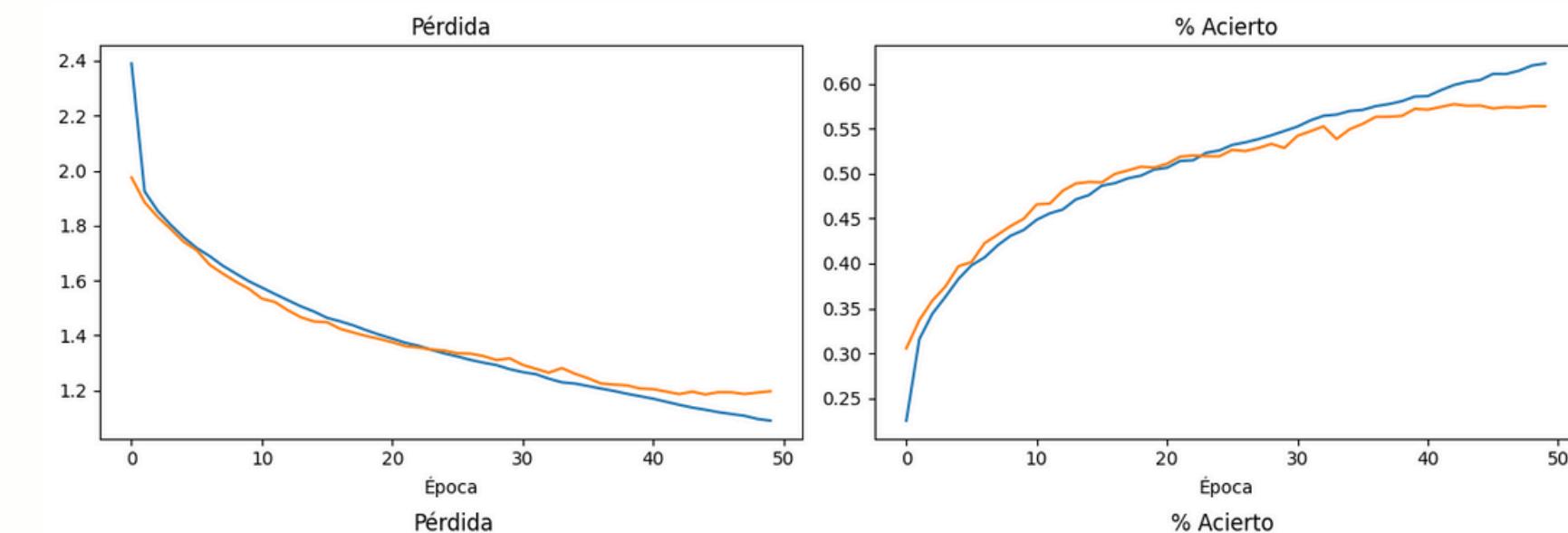
# SGD + Sigmoid

Capas	Sigmoid				
	accuracy	loss	val_accur acy	val_loss	tiempo
1	<b>0,4004</b>	<b>1,7536</b>	<b>0,3949</b>	<b>1,7515</b>	<b>59</b>
2	0,3269	1,9224	0,3330	1,9123	65
3	0,2126	2,1884	0,1909	2,1804	67



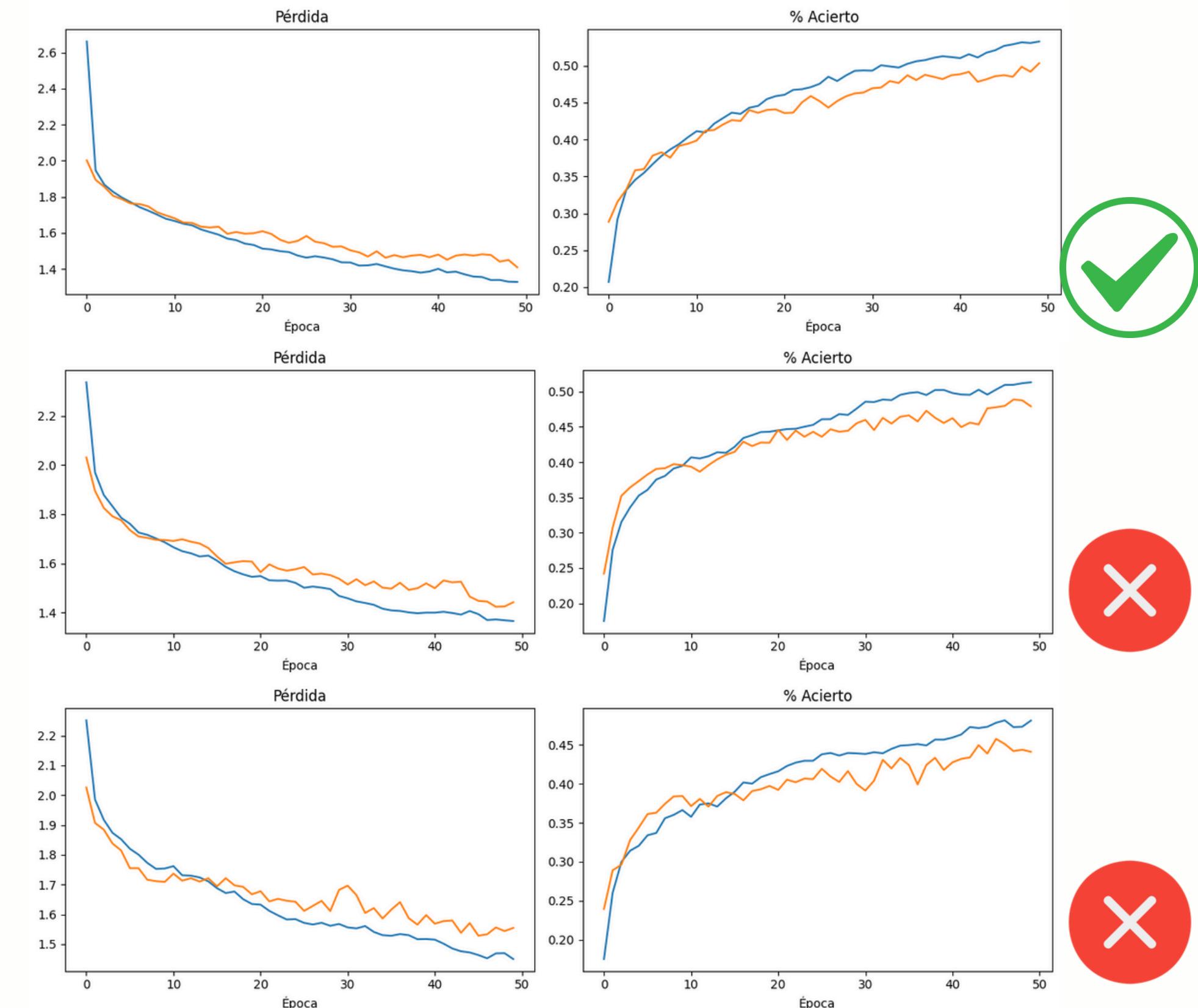
# ADAM + ReLu

Capas	Relu				
	accuracy	loss	val_accur acy	val_loss	tiempo
1	0,6154	1,1054	0,5748	1,1962	174
2	0,6713	0,9236	0,5989	1,1495	180
3	<b>0,7160</b>	<b>0,7959</b>	<b>0,6039</b>	<b>1,1658</b>	<b>299</b>



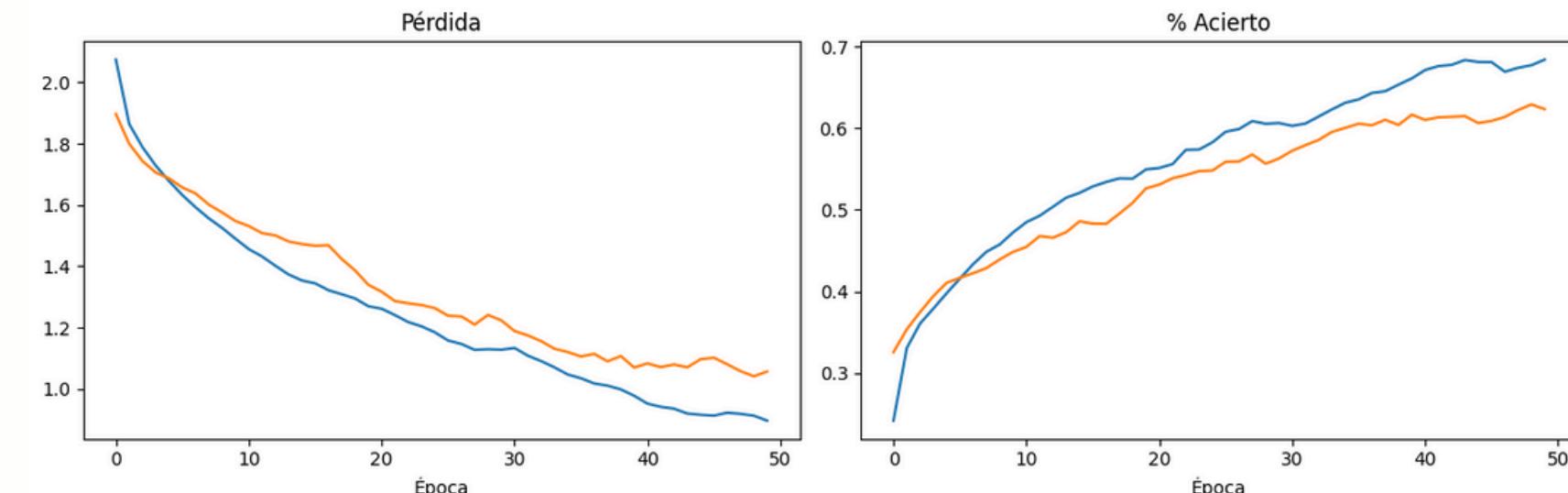
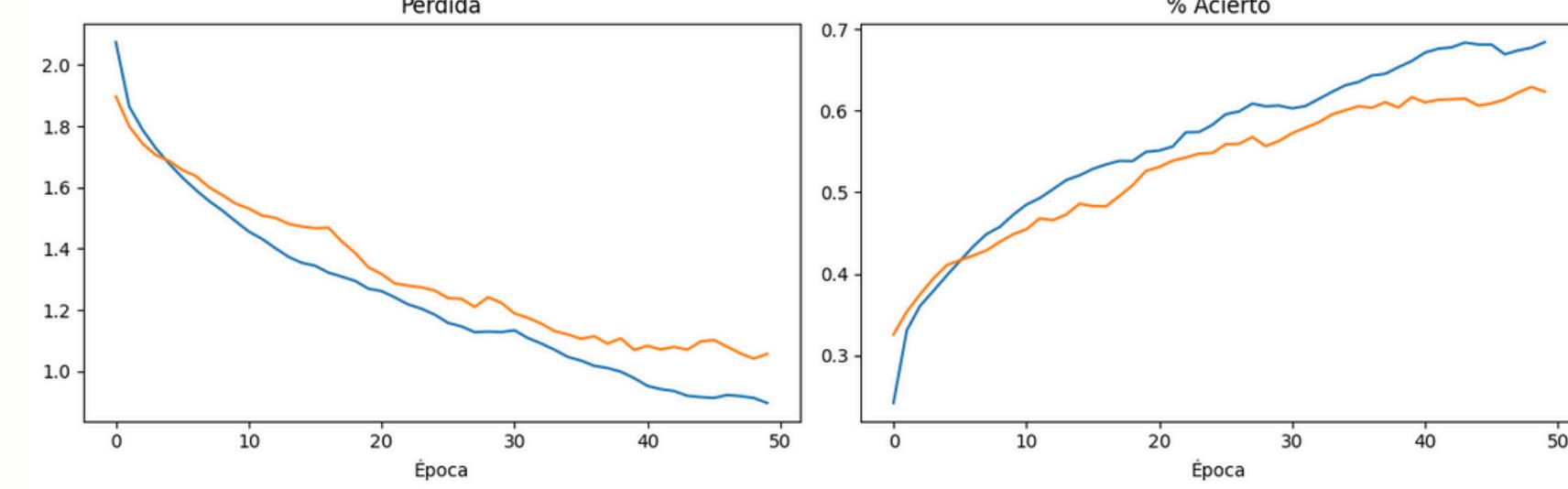
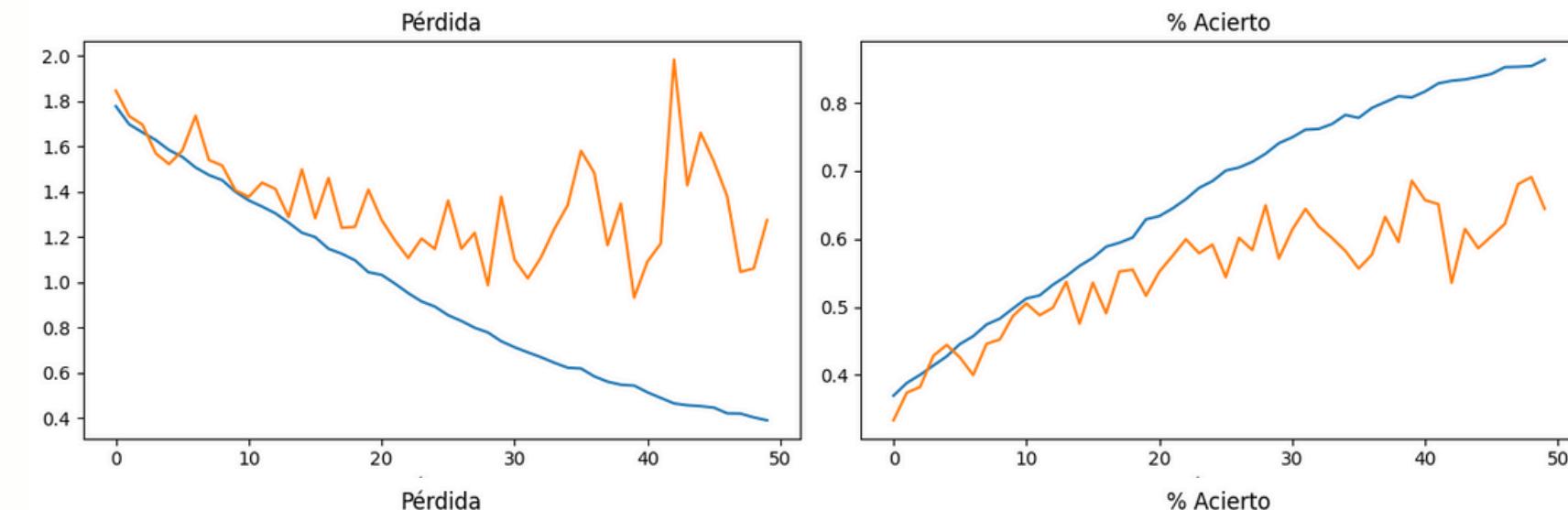
# ADAM + Tanh

Tanh					
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	<b>0,5350</b>	<b>1,3251</b>	<b>0,5034</b>	<b>1,4098</b>	<b>158</b>
2	0,5191	1,3531	0,4791	1,4417	183
3	0,4814	1,4582	0,4411	1,5545	193



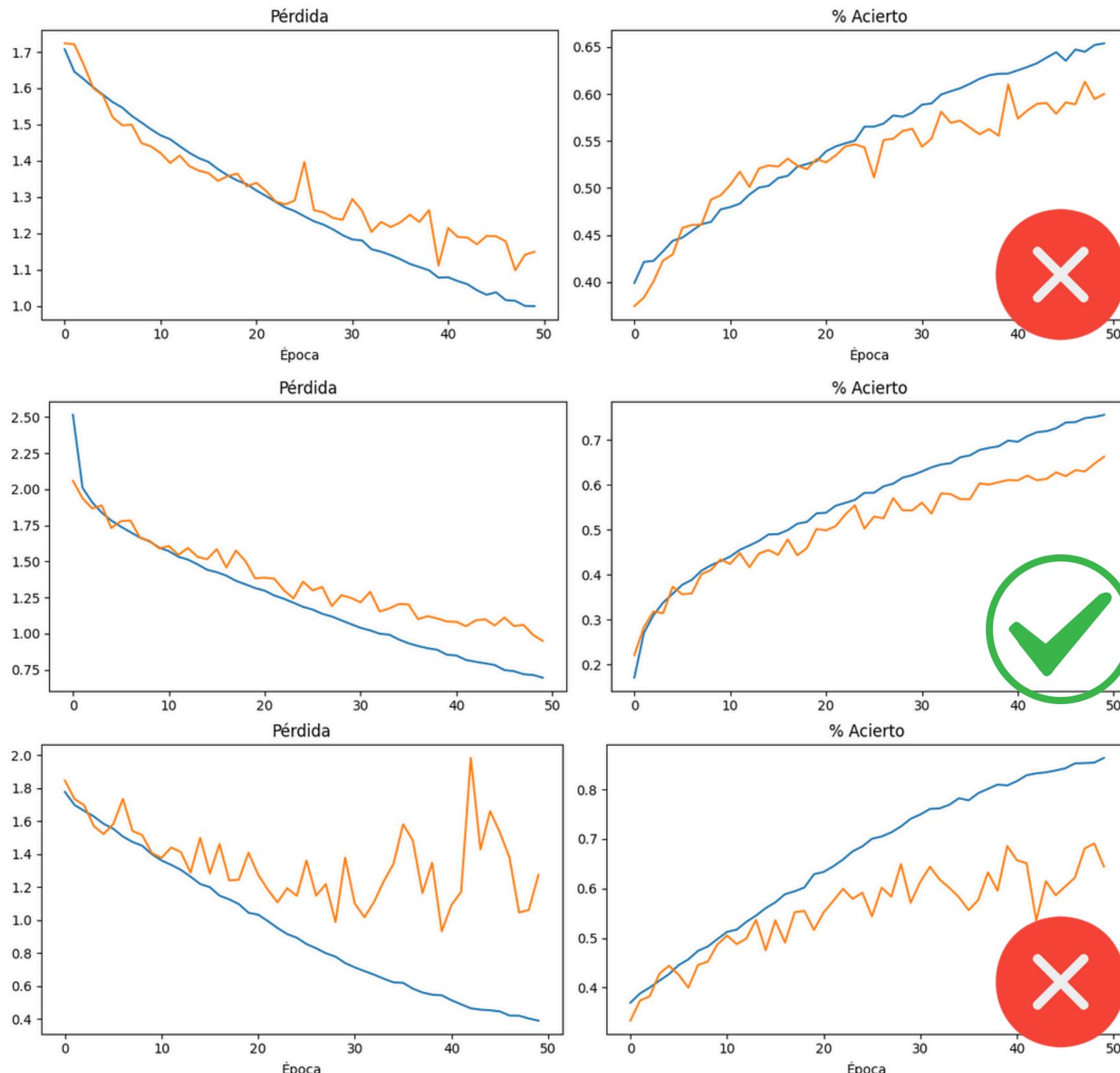
# ADAM + Sigmoid

	Sigmoid				
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	0,7428	0,8059	0,6807	0,9001	117
2	<b>0,7397</b>	<b>0,7860</b>	<b>0,6400</b>	<b>1,0510</b>	<b>114</b>
3	0,6813	0,9301	0,6490	0,9837	125



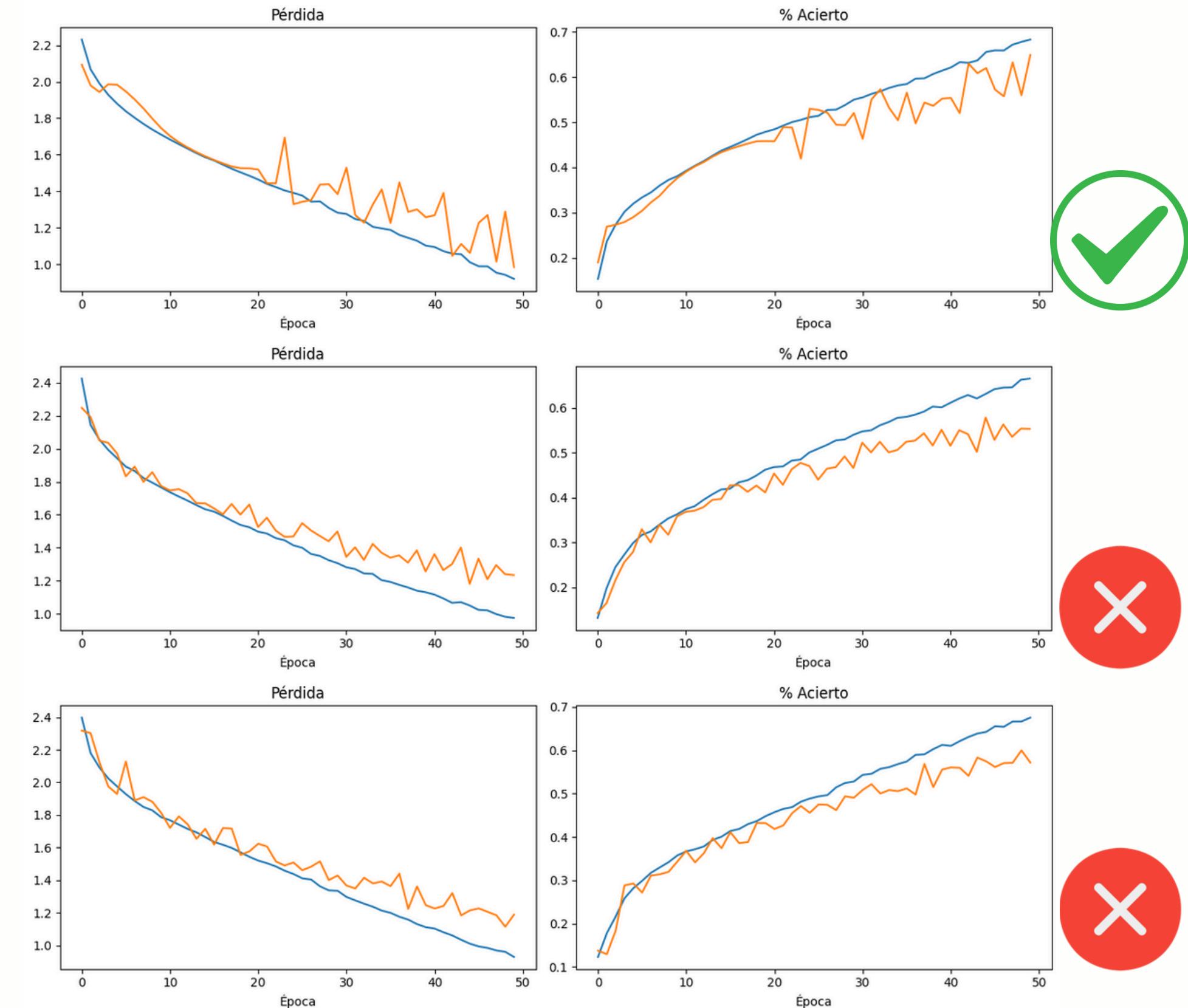
# RMSProp + ReLu

Relu					
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	0,6474	1,0088	0,6000	1,1492	119
2	<b>0,7576</b>	<b>0,6916</b>	<b>0,6630</b>	<b>0,9490</b>	<b>123</b>
3	0,8641	0,3901	0,6443	1,2746	132



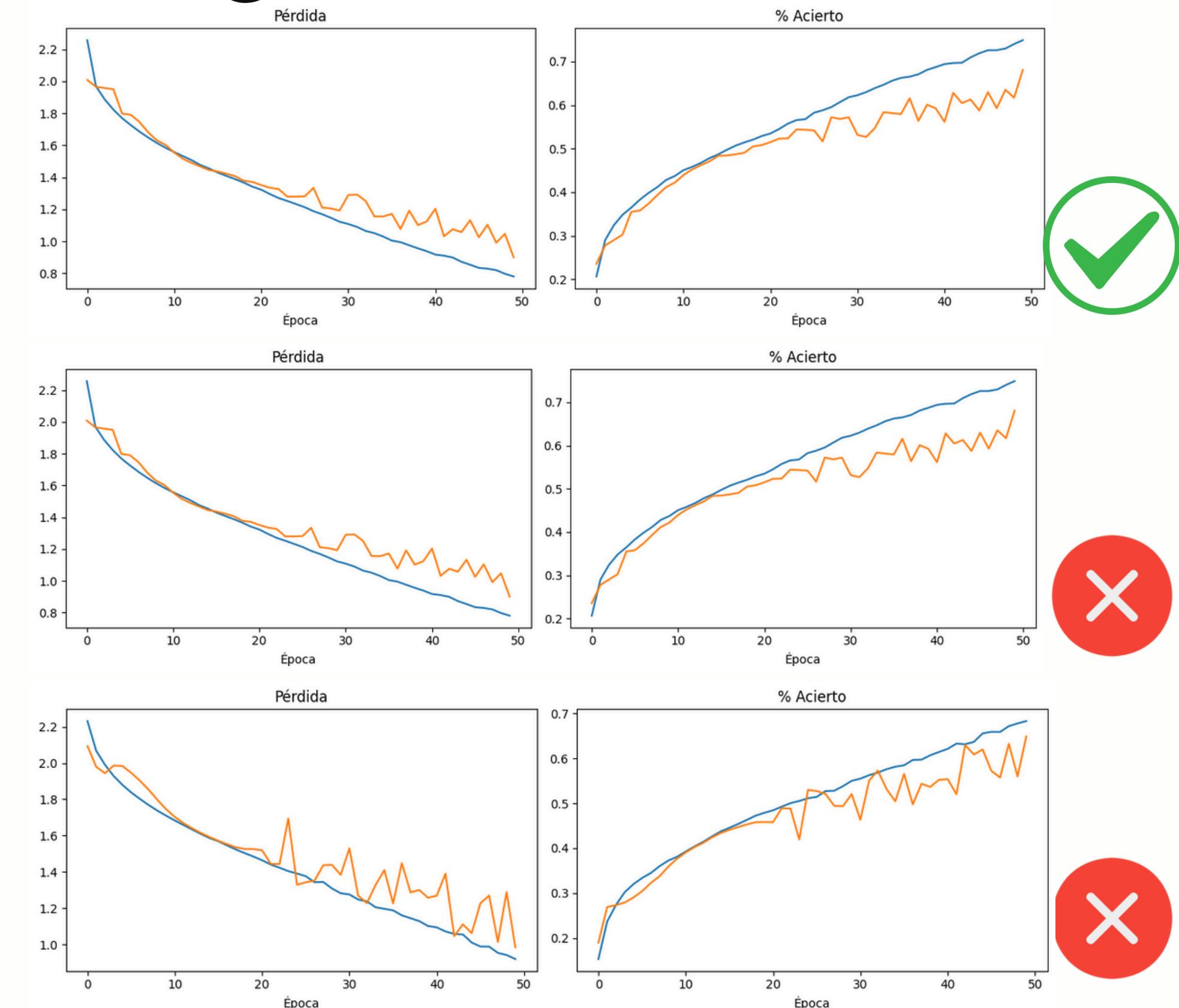
# RMSProp + Tanh

Tanh					
Capas	accuracy	loss	val_accur acy	val_loss	tiempo
1	<b>0,6218</b>	<b>1,108</b>	<b>0,4997</b>	<b>1,4631</b>	<b>117</b>
2	0,66	0,9886	0,5533	1,2350	126
3	0,6771	0,9359	0,5717	1,1889	129



# RMSProp + Sigmoid

Capas	Sigmoid				
	accuracy	loss	val_accur acy	val_loss	tiempo
1	<b>0,7428</b>	<b>0,8059</b>	<b>0,6807</b>	<b>0,9001</b>	<b>117</b>
2	0,7397	0,7860	0,6400	1,0510	114
3	0,6813	0,9301	0,6490	0,9837	125

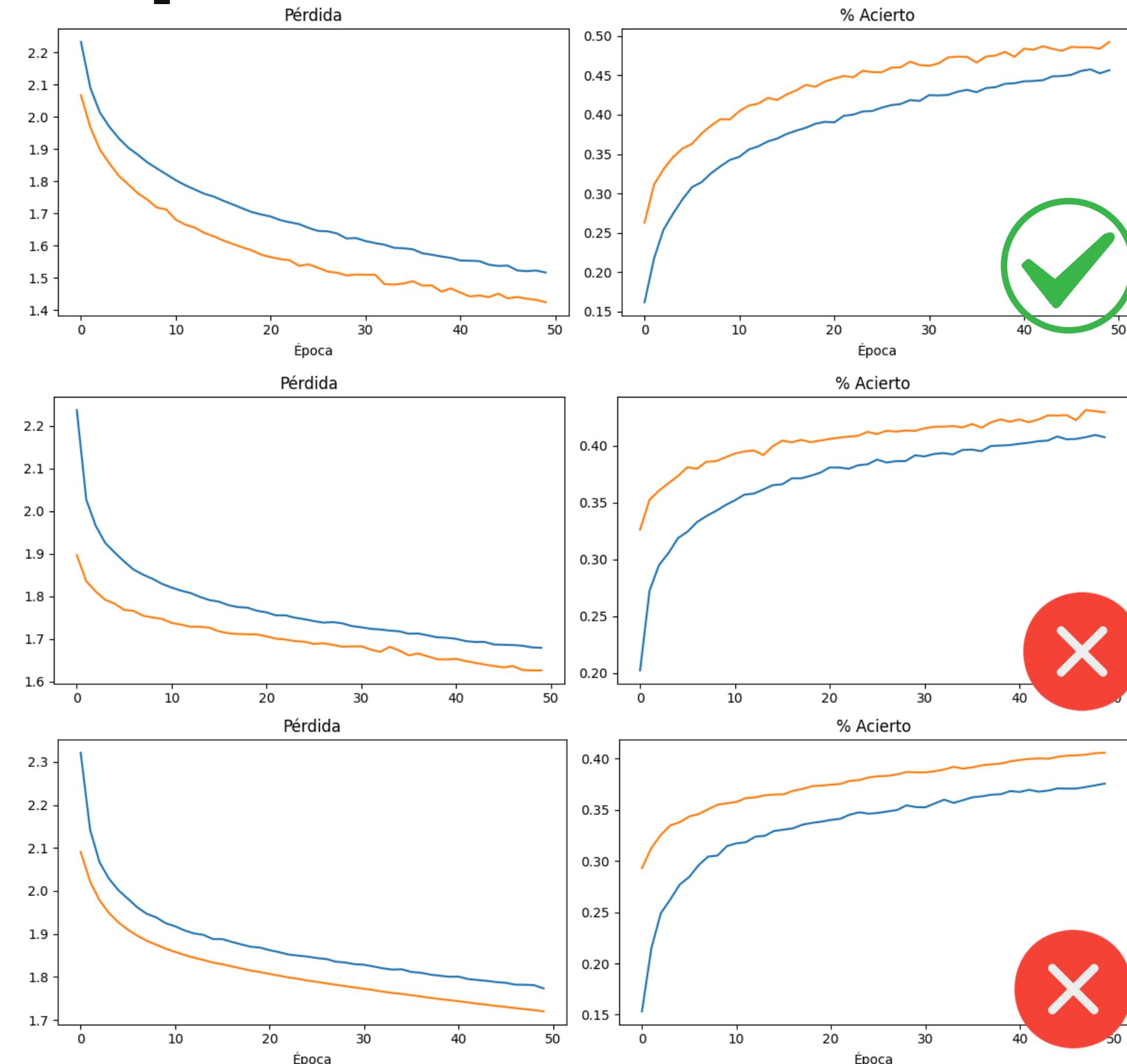


# Regularización

- Se escoge modelo con mejor rendimiento, tanto en SGD como en ADAM
  - Mejor combinación optimizador, función de activación y número de capas.
- Aplicamos Dropout y EarlyStopping

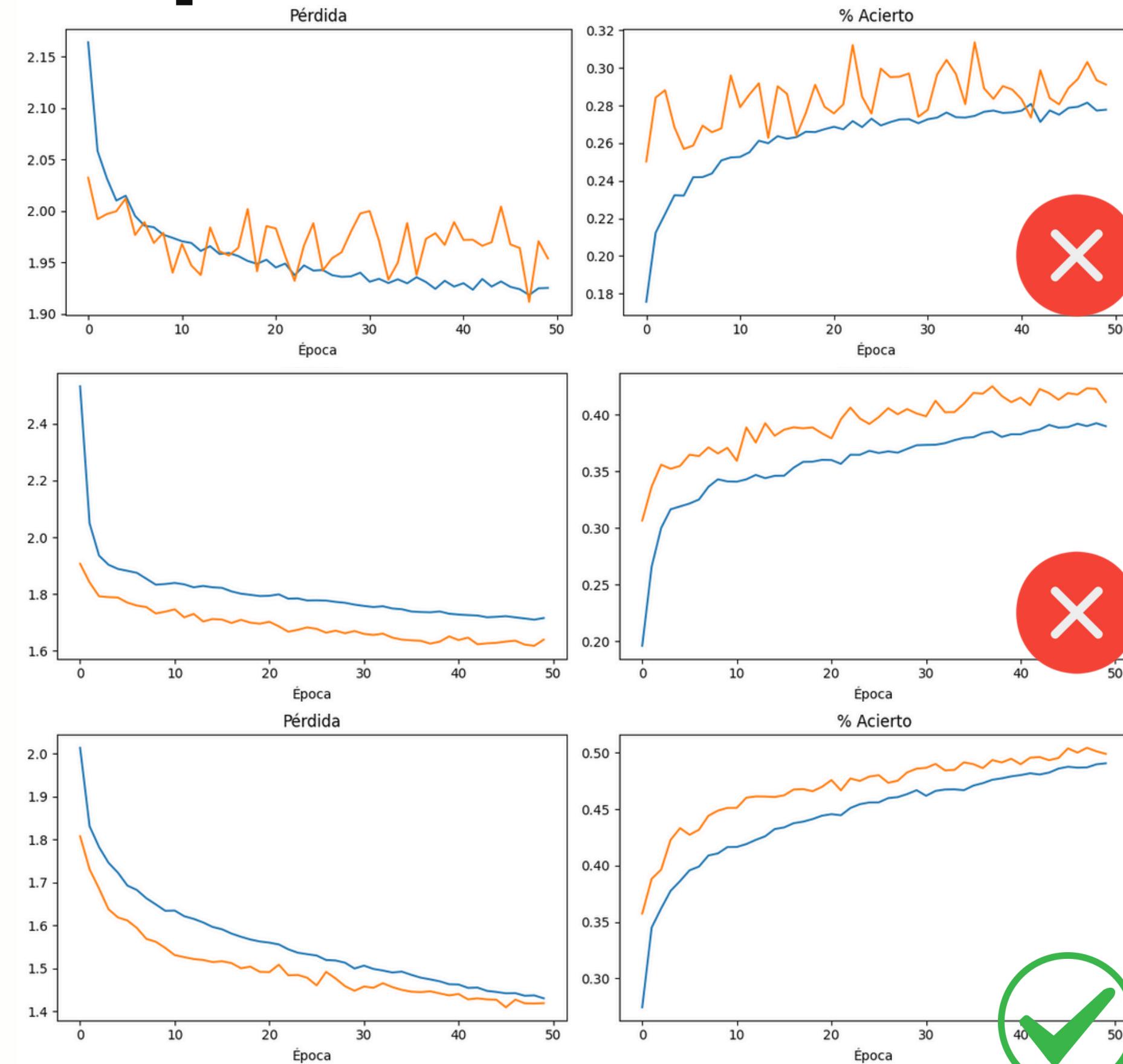
# SGD + Dropout

	Capas	accuracy	loss	val_accur acy	val_loss	tiempo
<b>Relu</b>	<b>3</b>	<b>0,4537</b>	<b>1,5168</b>	<b>0,4924</b>	<b>1,4245</b>	<b>247</b>
Tanh	3	0,4083	1,6801	0,4296	1,6265	244
Sigmoid	1	0,3705	1,7757	0,4058	1,7198	189



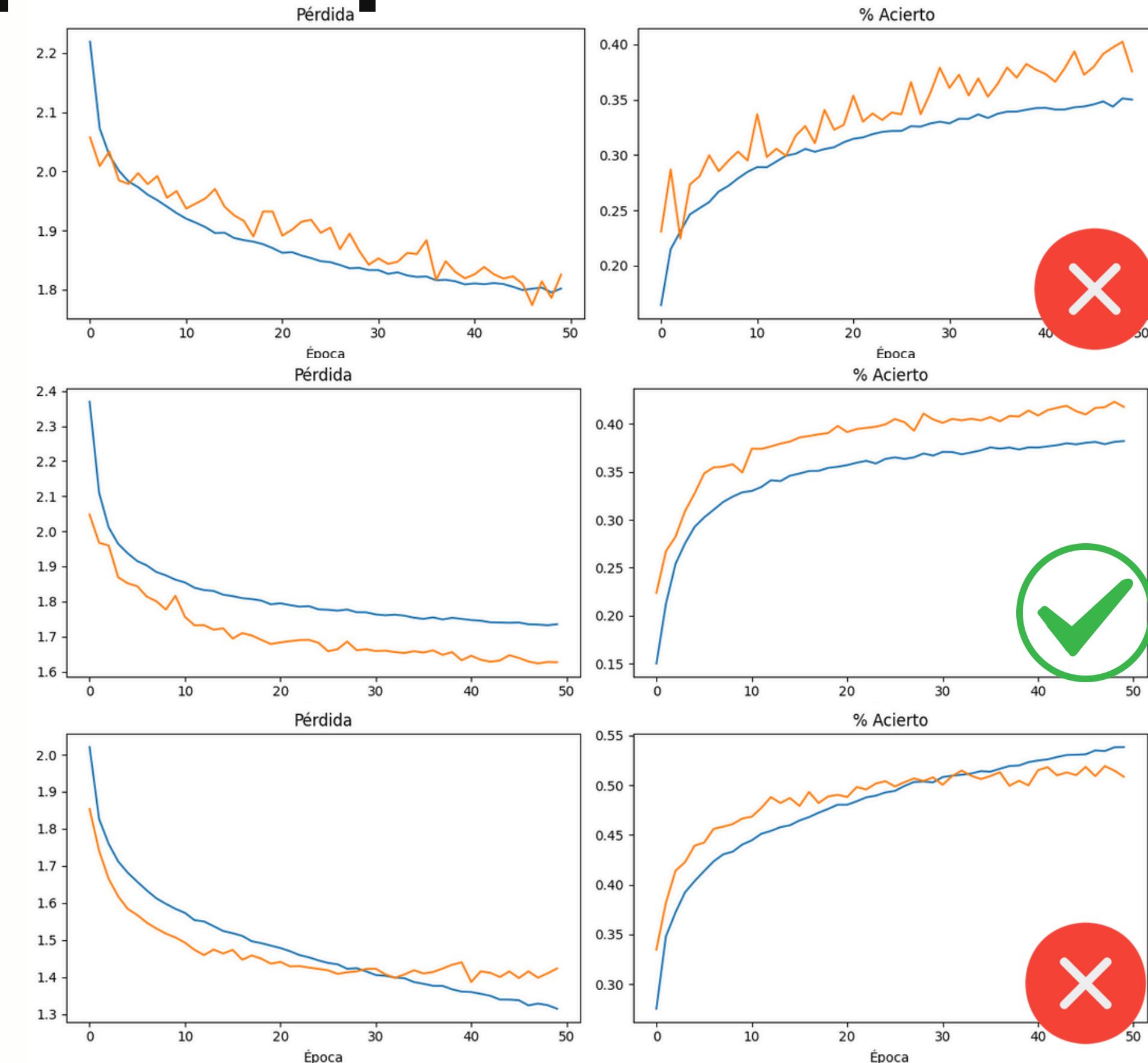
# ADAM + Dropout

	Capas	accuracy	loss	val_accuracy	val_loss	tiempo
Relu	3	0,2766	1,9304	0,2911	1,9537	501
Tanh	1	0,3905	1,7166	0,4114	1,6395	435
<b>Sigmoid</b>	<b>1</b>	<b>0,4919</b>	<b>1,4266</b>	<b>0,4991</b>	<b>1,4188</b>	<b>491</b>



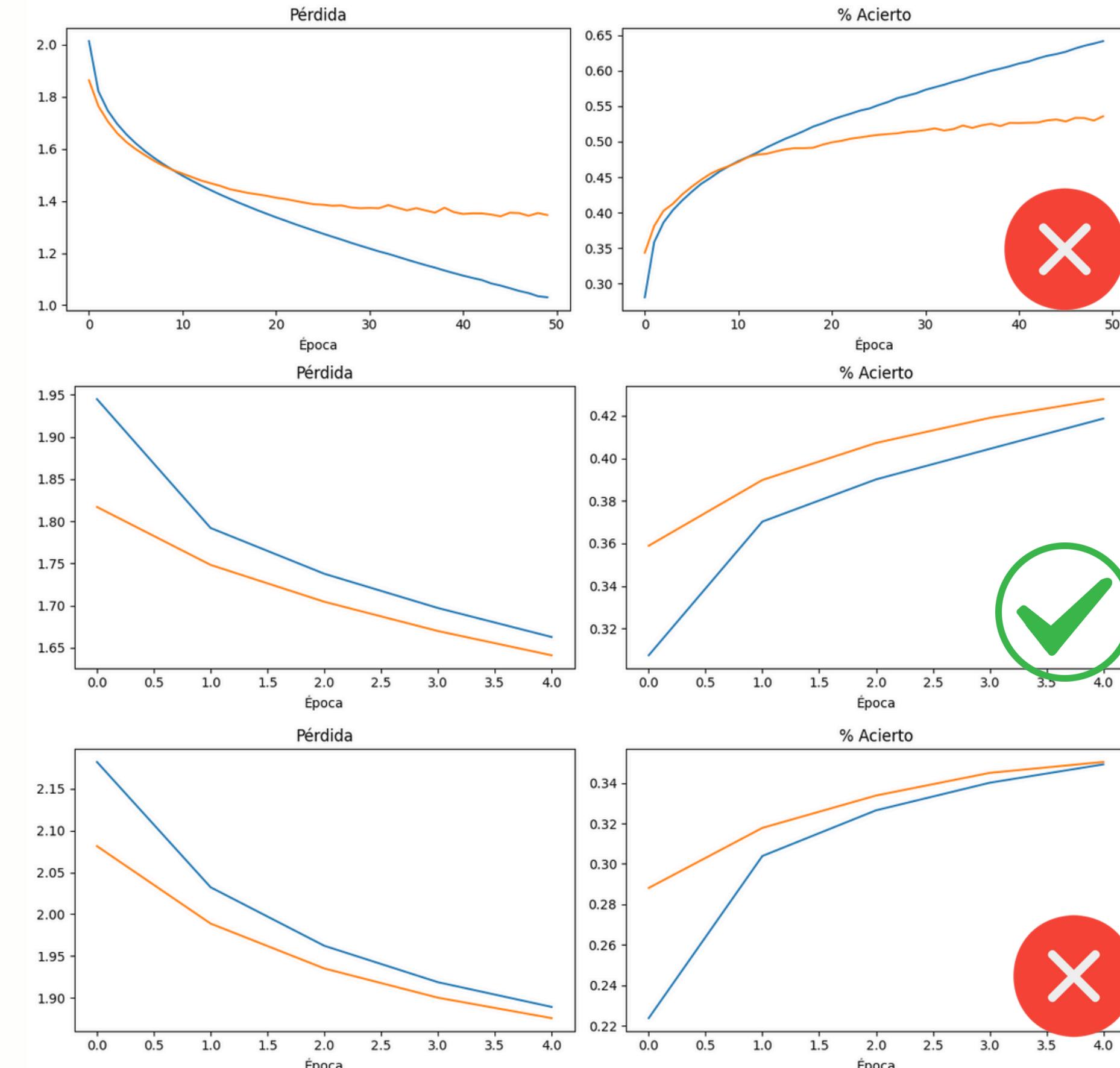
# RMSProp + Dropout

	<i>Capas</i>	<i>accuracy</i>	<i>loss</i>	<i>val_accuracy</i>	<i>val_loss</i>	<i>tiempo</i>
Relu	3	0,3462	1,8060	0,3756	1,8251	450
<b>Tanh</b>	<b>3</b>	<b>0,3835</b>	<b>1,7362</b>	<b>0,4178</b>	<b>1,6266</b>	<b>465</b>
Sigmoid	1	0,5405	1,3153	0,5084	1,4231	422



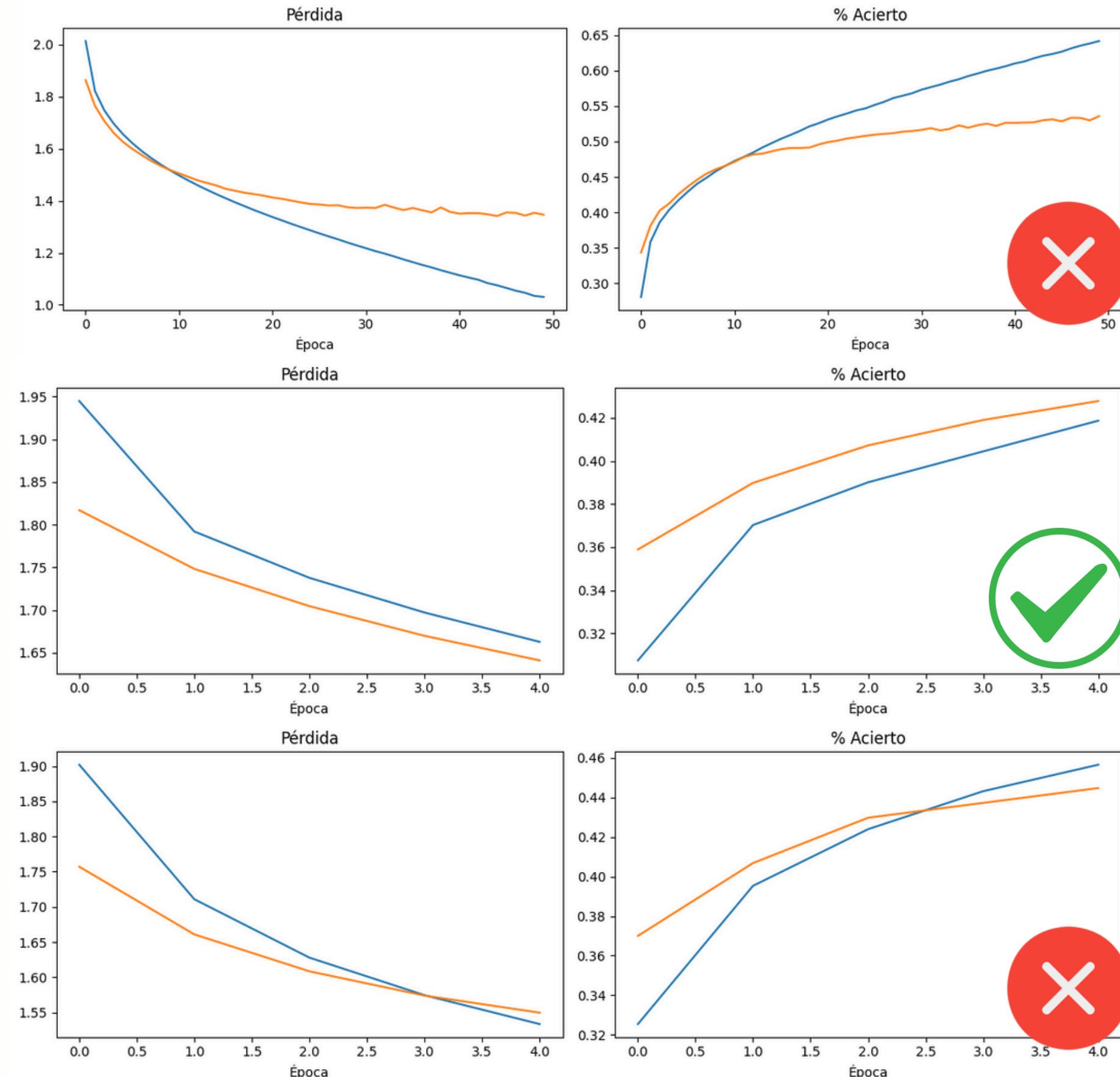
# SGD + Earlystopping

	Capas	acc	loss	val_acc	val_loss	tiempo	epoch
Relu	3	0,6413	1,0331	0,5357	1,3461	230	50
<b>Tanh</b>	<b>3</b>	<b>0,4171</b>	<b>1,6724</b>	<b>0,4278</b>	<b>1,6412</b>	<b>27</b>	<b>5</b>
Sigmoid	1	0,3473	1,8967	0,3504	1,8757	23	5



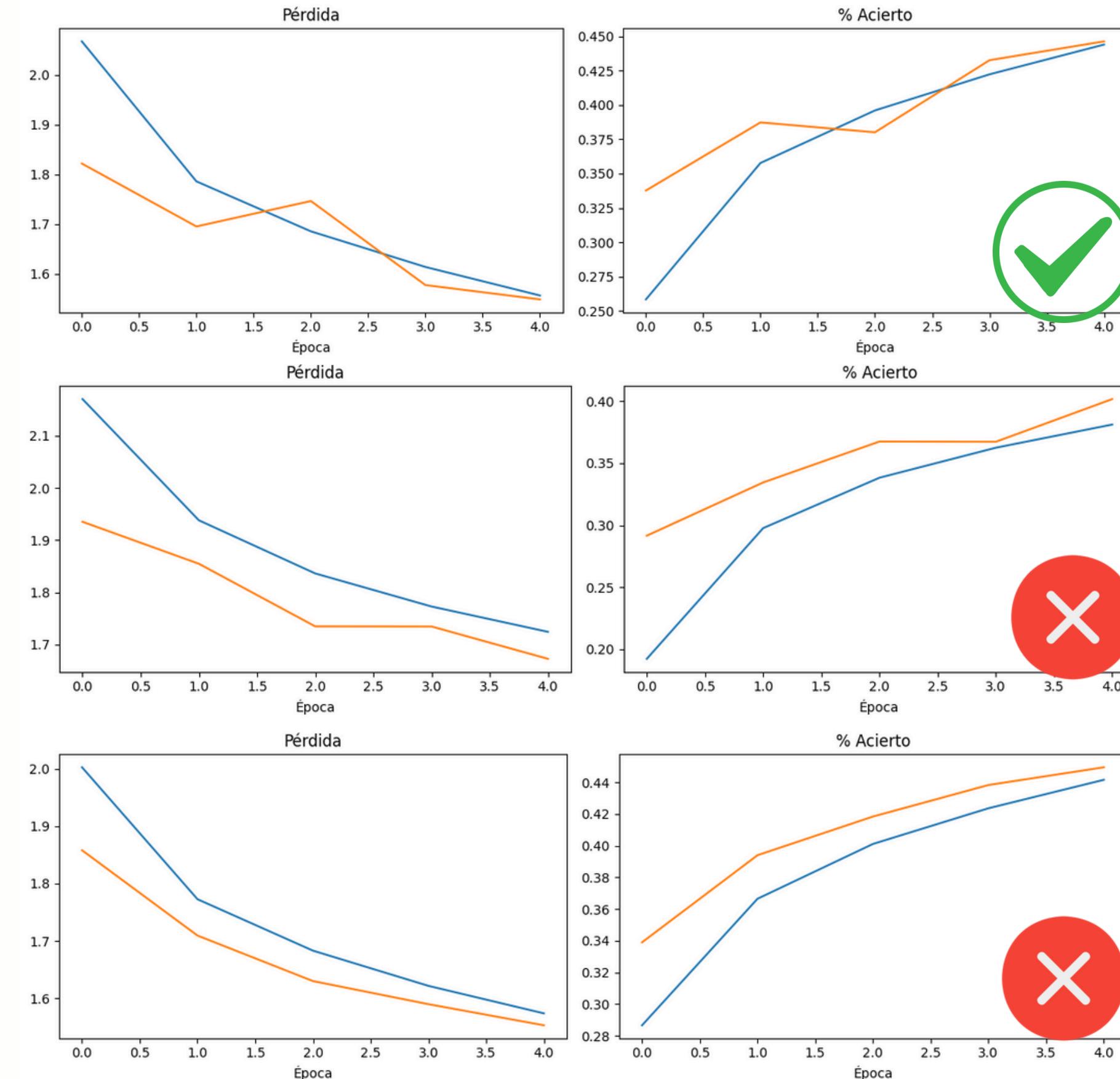
# ADAM + Earlystopping

	Capas	acc	loss	val_ac c	val_lo ss	tiempo	epoch
Relu	3	0,4620	1,4975	0,4621	1,5093	52	5
Tanh	1	<b>0,4096</b>	<b>1,6627</b>	<b>0,406 4</b>	<b>1,659 0</b>	<b>45</b>	<b>5</b>
Sigmoid	1	0,4522	1,5431	0,4447	1,5496	48	5



# RMSProp + Earlystopping

	Capas	acc	loss	val_acc	val_loss	tiempo	epoc
<b>Relu</b>	<b>3</b>	<b>0,436 4</b>	<b>1,5700</b>	<b>0,4463</b>	<b>1,5485</b>	<b>44</b>	<b>5</b>
Tanh	3	0,3706	1,7445	0,4017	1,6727	48	5
Sigmoid	1	0,4354	1,5872	0,4495	1,5596	44	5



# Evaluation:

- Más no siempre es mejor: Dependiendo la combinación, rendimiento mejora con menos capas.
- Regularizadores no mejora necesariamente accuracy pero sí el rendimiento de datos de prueba, especialmente con Sigmoid.
- La combinación SGD + ReLu mostró el mejor rendimiento, es decir, la mejor relación de accuracy entre los datos de entrenamiento y datos de prueba.



# Premios: (PARTE 1)

- Mejor optimizador: **SGD**
- Mejor función de activación: **ReLU**
- Mejor regularizador: **Dropout**
- Mejores combinaciones:
  - SGD: ReLu + 3 capas (60% vs 58%)
  - ADAM: ReLu + 3 capas (71% vs 60%)
  - RMSProp: Tanh + 3 capas (35% vs 41%)
  - Dropout: SGD + ReLu + 3 capas (45% vs 49%)
  - EarlyStopping: Adam + Tanh + 1 capa (40% vs 40%)



# Implementación Red Convolucional

Etiquetado de imágenes CIFAR-10

Para lo anterior debemos preparar nuestros datos tanto de entrenamiento como de prueba. Se normalizarán los datos dividiéndolos por 255 y nuestros datos de prueba se les aplicará el método `to_categorical`, que convertirá un vector de clase (enteros) en una matriz de clase binaria. Dicho de otra manera, se convertirán las etiquetas a one-hot encoding. Esto será útil para su uso con `categorical_crossentropy`.

.....

# Resultados Red Convolucional Personalizada

16:41 mins vs 57:07 mins

Como podemos observar, el rendimiento de nuestro modelo mejoró considerablemente, con una tasa de aprendizaje del 66% en la última época que el modelo siguió mejorando sus datos de aprendizaje, gracias al Early Stopping implementado. Hay que hacer presente que a pesar del Early Stopping, el tiempo de ejecución de nuestra red aumentó de 16 a 57 minutos.

Modelo	Capas	Épocas	Kernel	Tamaño Kernel	Padding	Pooling	Stride	Val-Acc	Val_loss	Tiempo (mins)
Base	4	25	Conv2d	3x3	valid	MaxPool2D	1	0.5817	2.5520	16:41
Mejorado	8	50	Conv2d	3x3	valid	MaxPool2D	1	0.6670	1.0771	57:07

# Resultados Red Convolucional: ResNet50

11:18 mins vs 51:34 mins

Los resultados nos enseñan que el modelo ha mejorado su capacidad de aprendizaje, aunque a un ritmo más lento (de 11 a 51 minutos). Es importante señalar que durante la ejecución es posible encontrar algunos warnings, los cuales significan que posiblemente no hay datos suficientes en nuestro dataset para cada época.

Modelo	Capas	Épocas	Kernel	Tamaño Kernel	Padding	Pooling	Stride	Val-Acc	Val_loss	Tiempo (mins)
Base	50	10	Conv2d (ResNet50)	3x3	same	GlobalAverage Pooling2D	2	0.3458	1.8575	11:18
Mejorado	50	50	Conv2d (ResNet50)	3x3	same	GlobalAverage Pooling2D	2	0.3458	1.8213	51:34

# Resultados Red Convolucional: Transfer Learning

50:55 mins vs 104 mins

El transfer learning (aprendizaje por transferencia) se basa en el concepto de reutilización de elementos de los modelos de machine learning (aprendizaje automático) preentrenados en nuevos modelos que se utilizarán para fines similares. Esto permite optimizar los recursos y los datos etiquetados necesarios para el entrenamiento

Modelo	Capas	Épocas	Kernel	Tamaño Kernel	Padding	Pooling	Stride	Val-Acc	Val_loss	Tiempo (mins)
Base	16	20	Conv2d (VGG16)	3x3	same	MaxPool2D	1	0.6158	2.1139	50:55
Mejorado	16	50	Conv2d (VGG16)	3x3	same	MaxPool2D	1	0.6409	1.0151	104

# Conclusiones:

Al comparar los modelos, observamos las siguientes tendencias:

- **Precisión y pérdida de validación:**
  - Todos los modelos mejorados superan a los modelos base en precisión de validación (Val\_Acc) y pérdida de validación (Val\_Loss), lo cual demuestra que las técnicas de mejora implementados a cada una de las arquitecturas planteadas han resultado ser efectivas.
  - Las mejoras son consistentes y significativas en todos los modelos, pasando de un promedio de Val\_Acc de 0.5817 a 0.6670 y reduciendo la Val\_Loss de aproximadamente un promedio de 2.5520 a 1.0771.

# Conclusiones:

Al comparar los modelos, observamos las siguientes tendencias:

- **Tiempo de entrenamiento:**
  - El modelo mejorado de la Red Convolucional personalizada requiere aproximadamente 57 minutos, que es un incremento alto comparado con su base, pero que mejoró considerablemente sus resultados.
  - El modelo mejorado de ResNet50 toma 51 minutos, lo cual es bastante eficiente dado su incremento en precisión y reducción de pérdida.
  - El modelo mejorado de VGG16 toma significativamente más tiempo (104 minutos), casi el doble de su versión base. Una manera de disminuir de manera rápida este tiempo es reduciendo las épocas, a costa eso sí de disminuir la precisión del modelo. Sin embargo, después de la época 10 la tasa de mejora es moderado, por lo que podría ser un límite aceptable a implementar.

# Mejor Modelo:

- Ambas VGG16 y la CNN personalizada dieron buenos resultados, pero la **CNN** personalizada mejorada es más eficiente, logrando los mismos resultados en menos tiempo y con menos recursos, convirtiéndola en el mejor modelo.



# ¡Gracias!

