

# CNNs

Adán González Rodríguez  
Sara Porto Álvarez

**Sobre el código:**

[Link al código](#)

# Estructura del Dataset:

```
# Cargar las imágenes de entrenamiento, validación y prueba
train_dir = "/home/estudiante/Escritorio/APAU_BIO/Tema8/cats_vs_dogs_small/train"
val_dir = "/home/estudiante/Escritorio/APAU_BIO/Tema8/cats_vs_dogs_small/validation"
test_dir = "/home/estudiante/Escritorio/APAU_BIO/Tema8/cats_vs_dogs_small/test"
```

# Red neuronal:

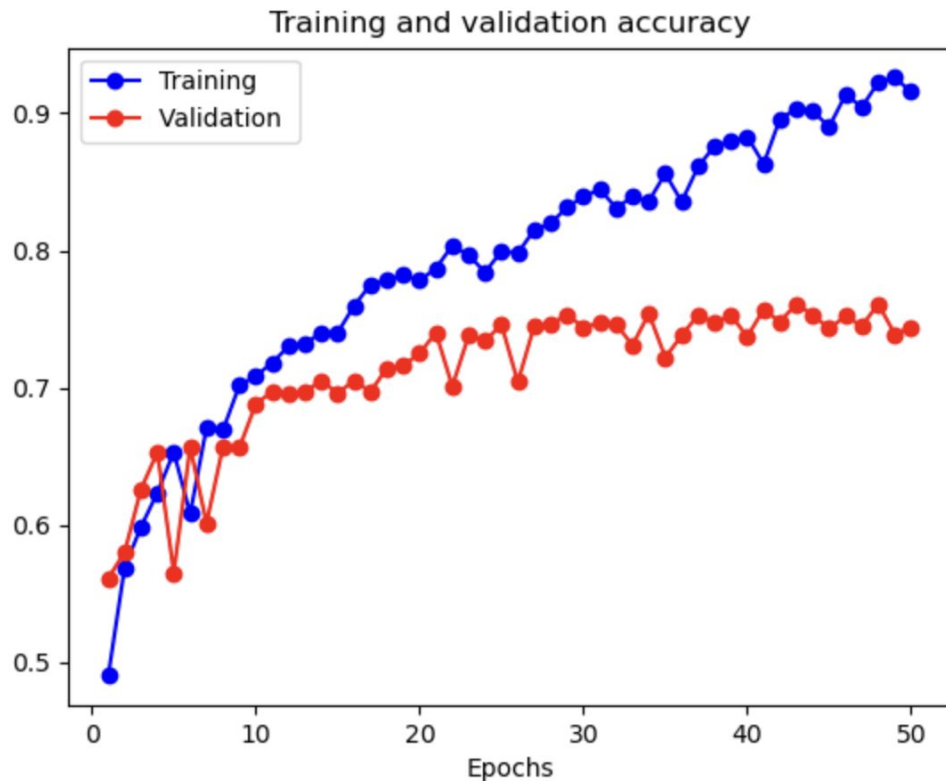
```
# Red neuronal simple
net = torch.nn.Sequential(
    torch.nn.Conv2d(3, 32, kernel_size=3),
    torch.nn.ReLU(),
    torch.nn.MaxPool2d(kernel_size=2),
    torch.nn.Conv2d(32, 64, kernel_size=3),
    torch.nn.ReLU(),
    torch.nn.MaxPool2d(kernel_size=2),
    torch.nn.Conv2d(64, 128, kernel_size=3),
    torch.nn.ReLU(),
    torch.nn.MaxPool2d(kernel_size=2),
    torch.nn.Conv2d(128, 256, kernel_size=3),
    torch.nn.ReLU(),
    torch.nn.MaxPool2d(kernel_size=2),
    torch.nn.Conv2d(256, 256, kernel_size=3),
    torch.nn.ReLU(),
    torch.nn.Flatten(),
    torch.nn.Linear(12544, num_classes),
    torch.nn.Softmax(dim=1)
).to(device)
```

# Resultados

# Resultados de una ejecución

```
Epoch 45: loss 0.4102 - accuracy 0.8105 - val. loss 0.0464 - val. acc. 0.6406  
Epoch 46: loss 0.4110 - accuracy 0.8135 - val. loss 0.0324 - val. acc. 0.7812  
Epoch 47: loss 0.4092 - accuracy 0.8145 - val. loss 0.0322 - val. acc. 0.7031  
Epoch 48: loss 0.4021 - accuracy 0.8205 - val. loss 0.0576 - val. acc. 0.5000  
Epoch 49: loss 0.4028 - accuracy 0.8160 - val. loss 0.0240 - val. acc. 0.8281  
Epoch 50: loss 0.3613 - accuracy 0.8445 - val. loss 0.0358 - val. acc. 0.7031
```

# Accuracy (los mejores encontrados)

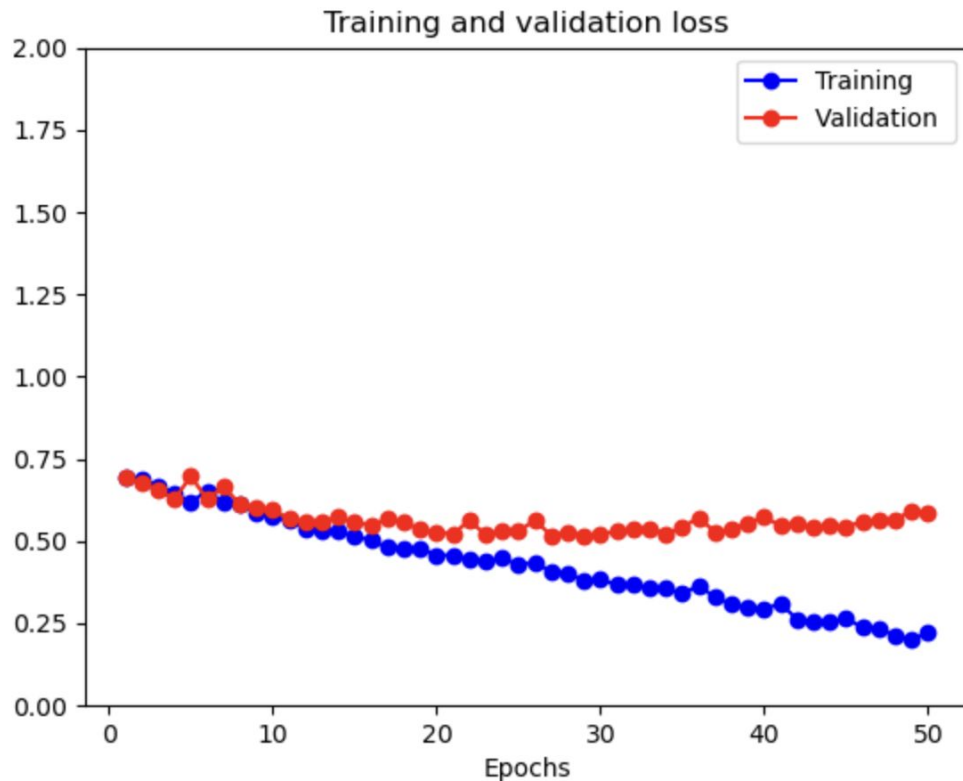


✅ **Precisión de entrenamiento alta:**  
+90% accuracy

⚠️ **Estancamiento en validación:** Se estabiliza sobre los 20 epochs

⚠️ **Sobreajuste (*overfitting*):** Gran diferencia entre train y validation

# Loss function (los mejores encontrados)



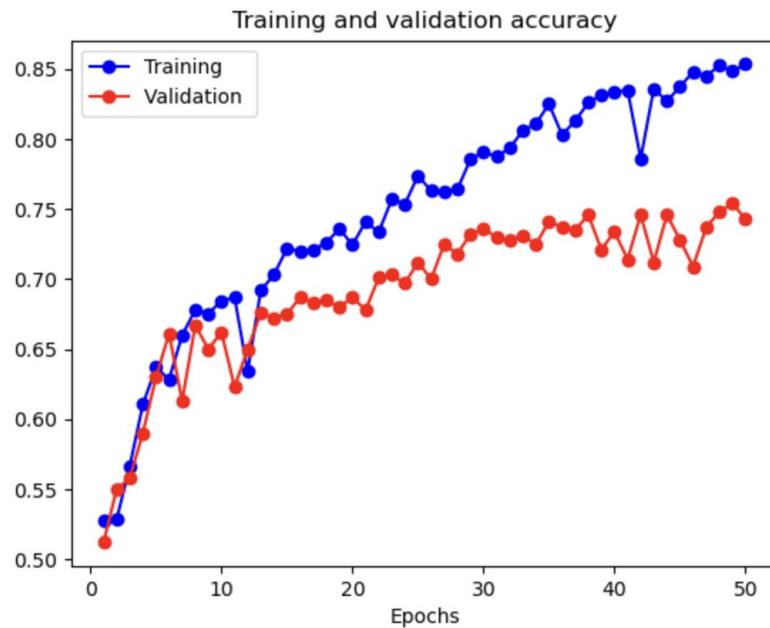
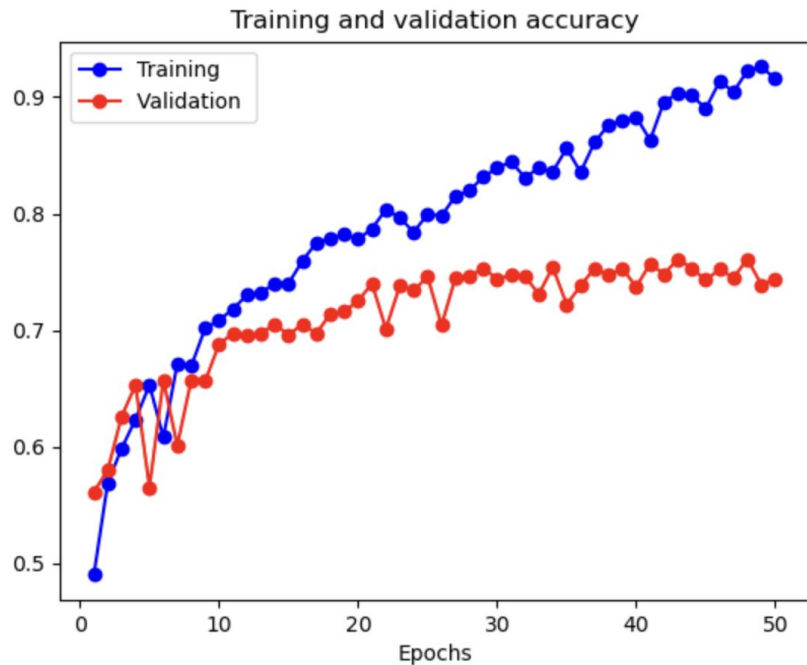
⚠ La pérdida de validación se estanca y luego sube ligeramente a partir de la época 20. Esto es una **señal clara de sobreajuste (overfitting)**

≡ Brecha entre las curvas → sobreajuste

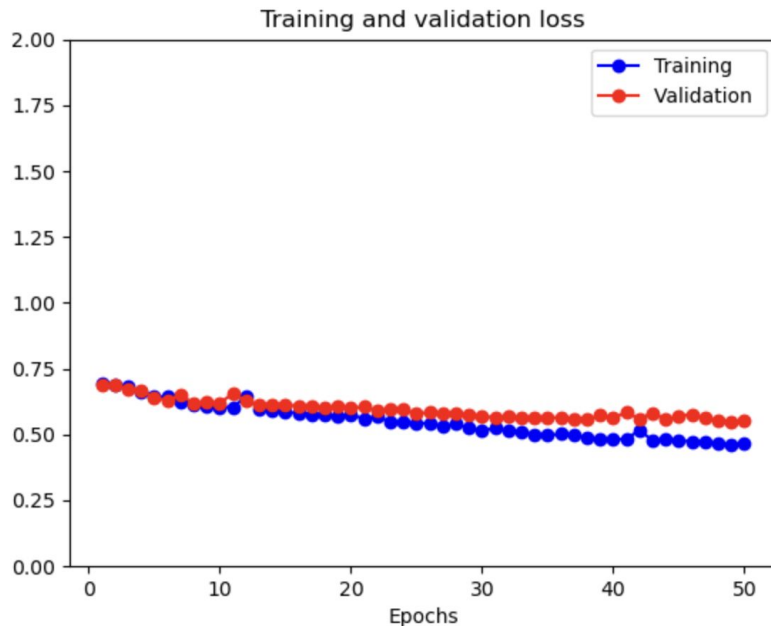
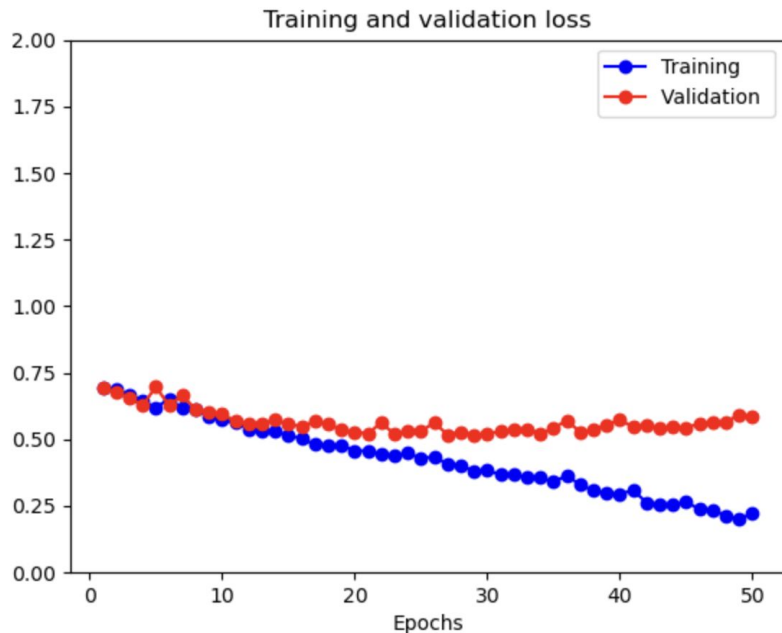
📊 Podría beneficiarse de **early stopping** (igual que *accuracy*)



# Accuracy sin y con data augmentation



# Loss function sin y con data augmentation



# Mosaico de imágenes sin y con data augmentation



**Resultados adicionales:**

[Link a resultados adicionales](#)