

Módulo 2. CNN: Reporte de resultados

José Benjamín Ruiz García

Descripción del problema

El problema que elegí es la clasificación de algunos medios de transporte y algunos animales. Las clases por identificar en este problema son:

0. Avión (airplane)
1. Carro (automobile)
2. Ave (bird)
3. Gato (cat)
4. Reno (deer)
5. Perro (dog)
6. Rana (frog)
7. Caballo (horse)
8. Barco (ship)
9. Camión (truck)

El dataset que usé para resolver este problema se llama *CIFAR10 small images classification dataset*, se compone por 60,000 imágenes de 32x32 píxeles a color. 50,000 imágenes son para el entrenamiento y 10,000 son para las pruebas.

Construcción del modelo

1

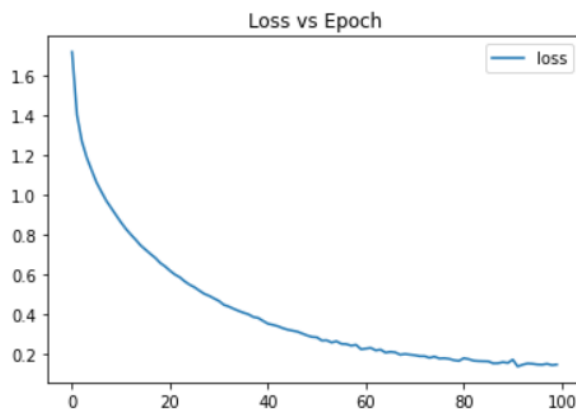
```
model = Sequential([
    Conv2D(32, kernel_size=3, padding='same', strides=2, activation="relu", input_shape=(32,32,3)),
    Conv2D(64, kernel_size=3, strides=1, activation='relu'),
    MaxPooling2D((3,3)),
    Conv2D(128, kernel_size=2, padding='same', strides=2, activation="relu"),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(10, activation='softmax')
])
```

Empecé con un modelo simple que tiene:

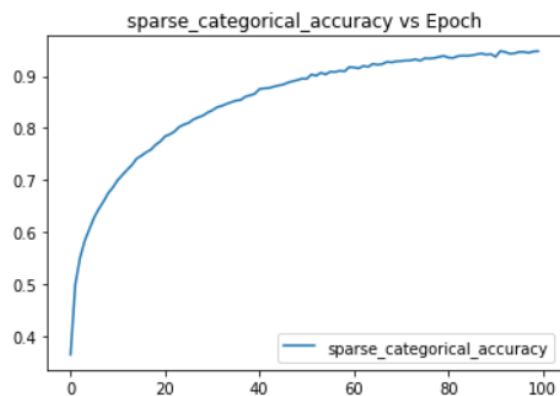
- Capa de convolución con 32 kernels de 3x3
- Capa de convolución con 64 kernels de 3x3
- Capa de maxpooling de 3x3
- Capa de convolución de 128 kernels de 2x2
- Capa de maxpooling de 2x2
- Capa de aplanamiento
- Capa densa con activación softmax con 10 neuronas (10 clases)

Para el entrenamiento utilicé el optimizador de Adam con un learning rate de 0.001, un batch size de 32 imágenes, 100 épocas y un 20% de datos para validación.

Resultados



Con el modelo 1 hubo una pérdida que llegó a 0.1, un valor muy cercano a 0 lo cual es prometedor.

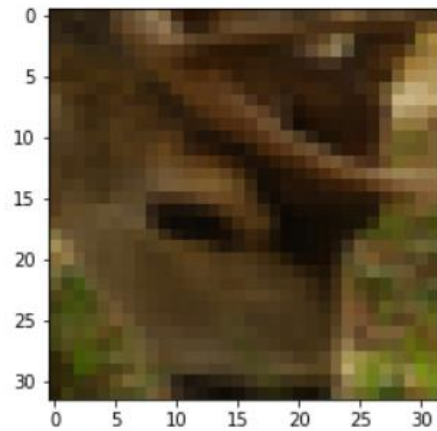


La curva de aprendizaje nos dice que el sparse categorical accuracy (sca) llegó a un valor muy cercano a 1 (0.94). Esto nos dice que el modelo aprendió bien, por lo que estamos listos para hacer pruebas.

Al momento de correr el modelo para evaluarlo con los datos de test los resultados fueron muy pobres:

```
loss 4.2496  
acc 0.58200
```

Hubo un sca de 0.58, lo cual está muy alejado de 1. Probé individualmente con una imagen al azar y la predicción fue errónea:



Label: deer

El modelo debió haber clasificado esto como un reno (deer). En vez de esto lo clasificó como una rana.

1/1 [=====] - 0s 17ms/step
Model frog

Optimización del modelo

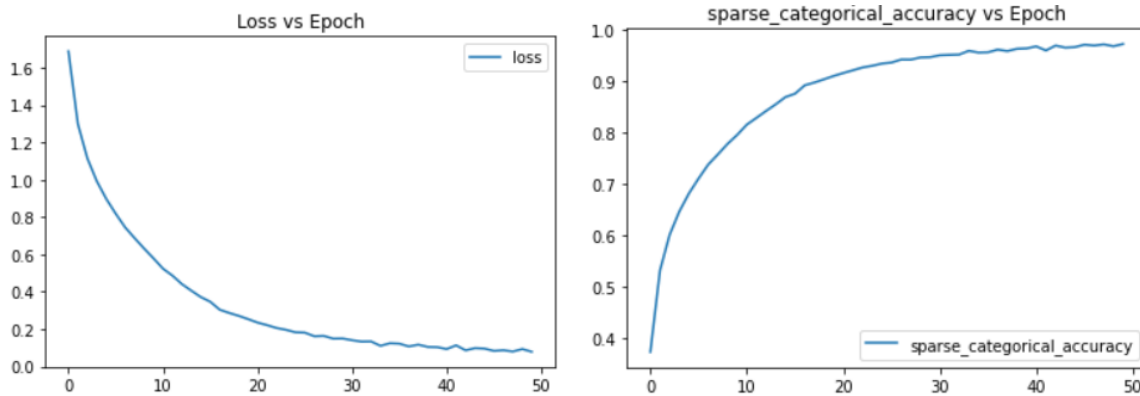
Para tratar de mejorar los resultados probé con arquitecturas diferentes variando hiperparámetros y llegué a esta:

- Capa de convolución con 200 kernels de 3x3
- Capa de convolución con 200 kernels de 3x3
- Capa de maxpooling de 2x2
- Capa de convolución con 160 kernels de 3x3
- Capa de maxpooling de 2x2
- Capa de aplanamiento
- Capa densa con 160 neuronas y función de activación relu
- Capa densa con 80 neuronas y función de activación relu
- Capa densa de 40 neuronas y función de activación relu
- Capa densa de 10 neuronas (10 clases) con función de activación softmax

```
model = Sequential([
    Conv2D(200, kernel_size=3, padding='same', strides=2, activation="relu", input_shape=(32,32,3)),
    Conv2D(200, kernel_size=3, padding='same', strides=2, activation="relu"),
    MaxPooling2D((2,2)),
    Conv2D(160, kernel_size=3, padding='same', strides=2, activation="relu"),
    MaxPooling2D((2,2)),
    Flatten(),
    Dense(160, activation='relu'),
    Dense(80, activation='relu'),
    Dense(40, activation='relu'),
    Dense(10, activation='softmax')
])
```

Para el entrenamiento utilicé el optimizador de Adam con un learning rate de 0.001, un batch size de 64 imágenes, 50 épocas y un 10% de datos para validación.

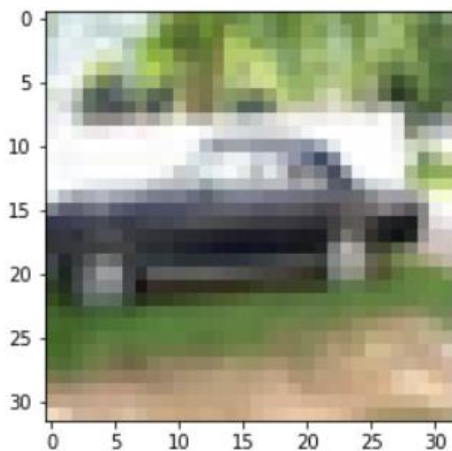
Esta configuración fue la que mejores resultados me dio:



```
loss 2.6296
acc 0.64499
```

Como se puede observar, a pesar de haber subido el sca a 0.64, los resultados siguen siendo pobres. Para mejorarlo podrían probarse otras arquitecturas más robustas que arrojen más feature maps de las imágenes para que el modelo aprenda de mejor manera.

Realicé una predicción para probar este modelo y obtuve lo siguiente:



Label: automobile

En este caso el modelo sí clasificó esta imagen como un automóvil lo cual demuestra una mejora en los resultados que obtuvimos anteriormente.

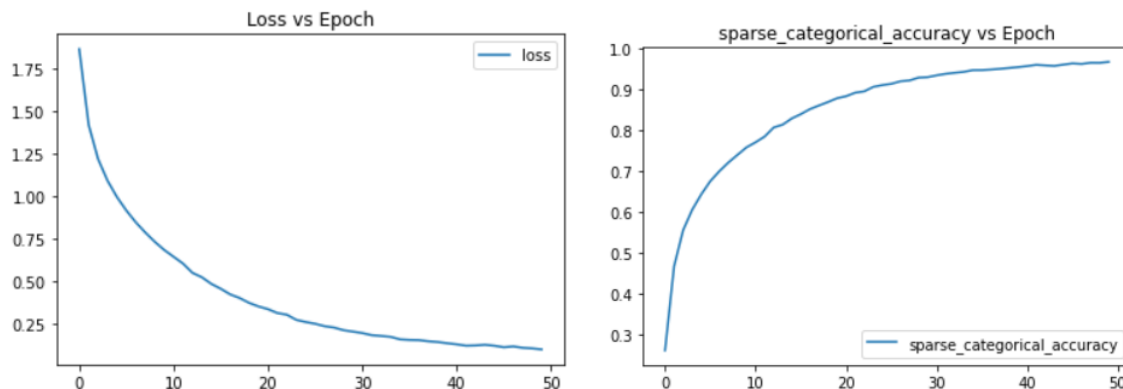
```
1/1 [=====] - 0s 249ms/step
Model automobile
```

Por último, tomando en cuenta los comentarios del profesor, cambié la arquitectura del modelo. Esta vez puse kernels de manera incremental, en cada capa de convolución subí el número de kernels ya que de esta manera es como se acostumbra a hacerlo. La red quedó de la siguiente manera:

- Capa de convolución con 50 kernels de 3x3
- Capa de convolución con 100 kernels de 3x3
- Capa de maxpooling de 2x2
- Capa de convolución con 200 kernels de 3x3
- Capa de maxpooling de 2x2
- Capa de convolución con 300 kernels de 3x3
- Capa de convolución con 500 kernels de 3x3
- Capa de aplanamiento
- Capa densa con 500 neuronas y función de activación relu
- Capa densa con 300 neuronas y función de activación relu
- Capa densa con 100 neuronas y función de activación relu
- Capa densa de 10 neuronas (10 clases) con función de activación softmax

```
1 model = Sequential([
2     Conv2D(50, kernel_size=3, padding='same', strides=2, activation="relu", input_shape=(32,32,3)),
3     Conv2D(100, kernel_size=3, padding='same', strides=2, activation="relu"),
4     MaxPooling2D((2,2)),
5     Conv2D(200, kernel_size=3, padding='same', strides=2, activation="relu"),
6     MaxPooling2D((2,2)),
7     Conv2D(300, kernel_size=3, padding='same', strides=2, activation="relu"),
8     Conv2D(500, kernel_size=3, padding='same', strides=2, activation="relu"),
9     Flatten(),
10    Dense(500, activation='relu'),
11    Dense(300, activation='relu'),
12    Dense(100, activation='relu'),
13    Dense(10, activation='softmax')
14 ])
```

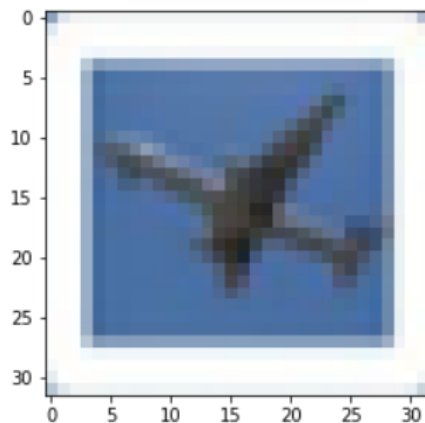
Para el entrenamiento utilicé el optimizador de Adam con un learning rate de 0.001, un batch size de 64 imágenes, 50 épocas y un 10% de datos para validación. Con esto obtuve los siguientes resultados:



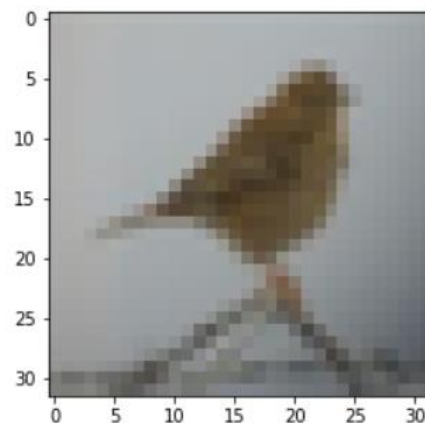
loss 2.604
acc 0.6374

Como podemos observar el sca casi no cambió con respecto al sca de la arquitectura anterior (0.64). Para mejorarlo tal vez sacando más feature maps de las imágenes y utilizando otra capa de maxpooling podría subir el sca.

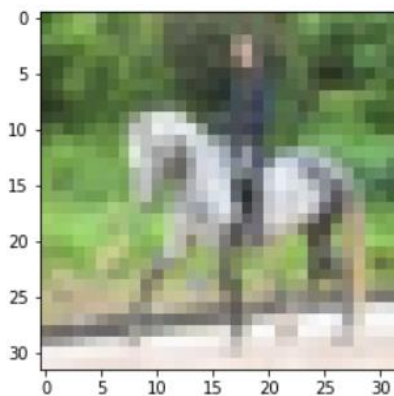
A pesar de que los resultados no fueron los mejores, estuve haciendo pruebas y el modelo sí predice bien las clases la mayoría de las veces:



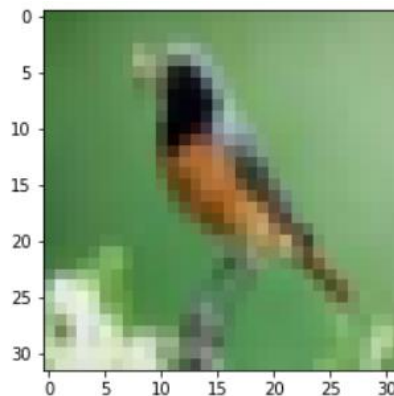
Prediction: airplane



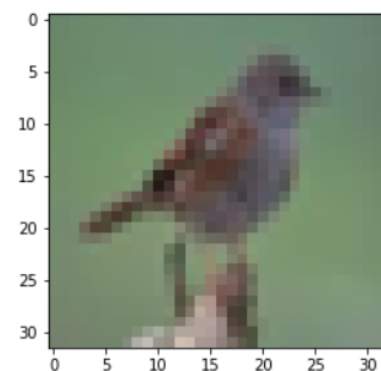
Prediction: bird



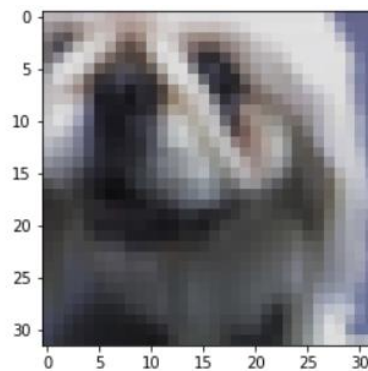
Prediction: horse



Prediction: bird



Prediction: bird



Prediction: dog