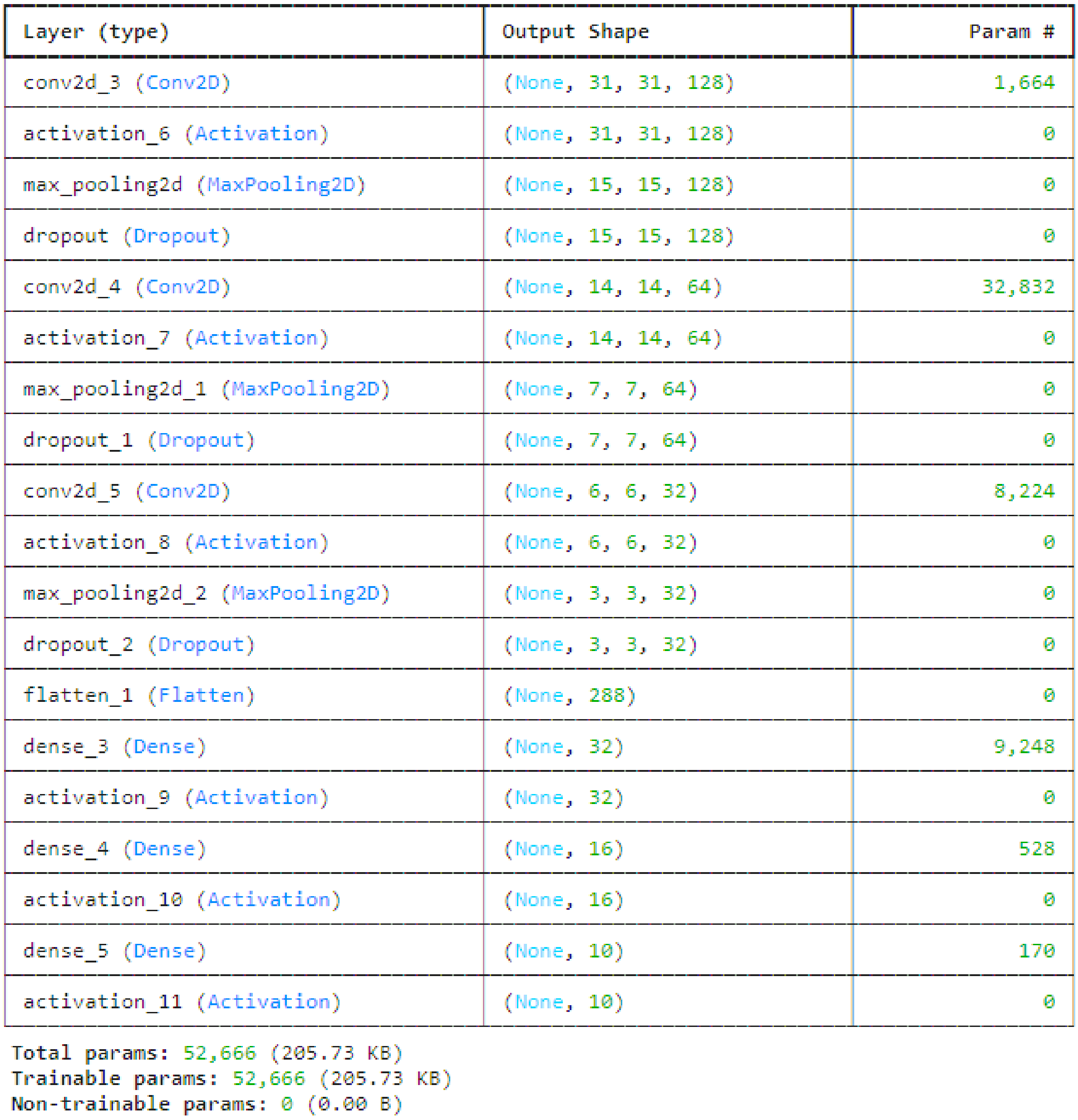
Filager MLOps School

EXC 13 | Abolfazl Aghdaee

In this practice I implemented a Deep Neural Network for classifying the CIFAR-10 dataset.

Model architecture:

Our model contains Convolutions, MaxPoolings, Flatten, and Dropout layers. I used Relu and Softmax as activation functions.
A brief summary of the model is represented as follows:



Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 31, 31, 128)	1,664
activation_6 (Activation)	(None, 31, 31, 128)	0
max_pooling2d (MaxPooling2D)	(None, 15, 15, 128)	0
dropout (Dropout)	(None, 15, 15, 128)	0
conv2d_4 (Conv2D)	(None, 14, 14, 64)	32,832
activation_7 (Activation)	(None, 14, 14, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
dropout_1 (Dropout)	(None, 7, 7, 64)	0
conv2d_5 (Conv2D)	(None, 6, 6, 32)	8,224
activation_8 (Activation)	(None, 6, 6, 32)	0
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 32)	0
dropout_2 (Dropout)	(None, 3, 3, 32)	0
flatten_1 (Flatten)	(None, 288)	0
dense_3 (Dense)	(None, 32)	9,248
activation_9 (Activation)	(None, 32)	0
dense_4 (Dense)	(None, 16)	528
activation_10 (Activation)	(None, 16)	0
dense_5 (Dense)	(None, 10)	170
activation_11 (Activation)	(None, 10)	0

```
Total params: 52,666 (205.73 KB)
Trainable params: 52,666 (205.73 KB)
Non-trainable params: 0 (0.00 B)
```

Challenges:

While training the model, I encountered overfitting, so I introduced Dropout layers to mitigate it effectively.

I used the MaxPooling layers to improve the accuracy on training and validation data.

Comparison results:

A brief summary about model size, inference time, and accuracy as follows:

h5 model:

Model size:

```
h5_model size is :0.67 Mb
```

Inference Time:

0.9444911479949951

Accuracy:

0.7215

TFLITE model:

Model size:

```
tilite_model size is :0.21 Mb
```

Inference Time:

```
0.04320096969604492
```

Accuracy:

0.7215

ONNX model:

Model size:

```
onnx_model size is :0.20 Mb
```

Inference Time:

0.00800943374633789

Accuracy:

0.7215

As you observed, the accuracy is the same across all models; however, the inference time and model size of the .onnx model are superior compared to the other two models.

Suggestion:

We can use more convolution layers and more epochs for training to improve the model performance.