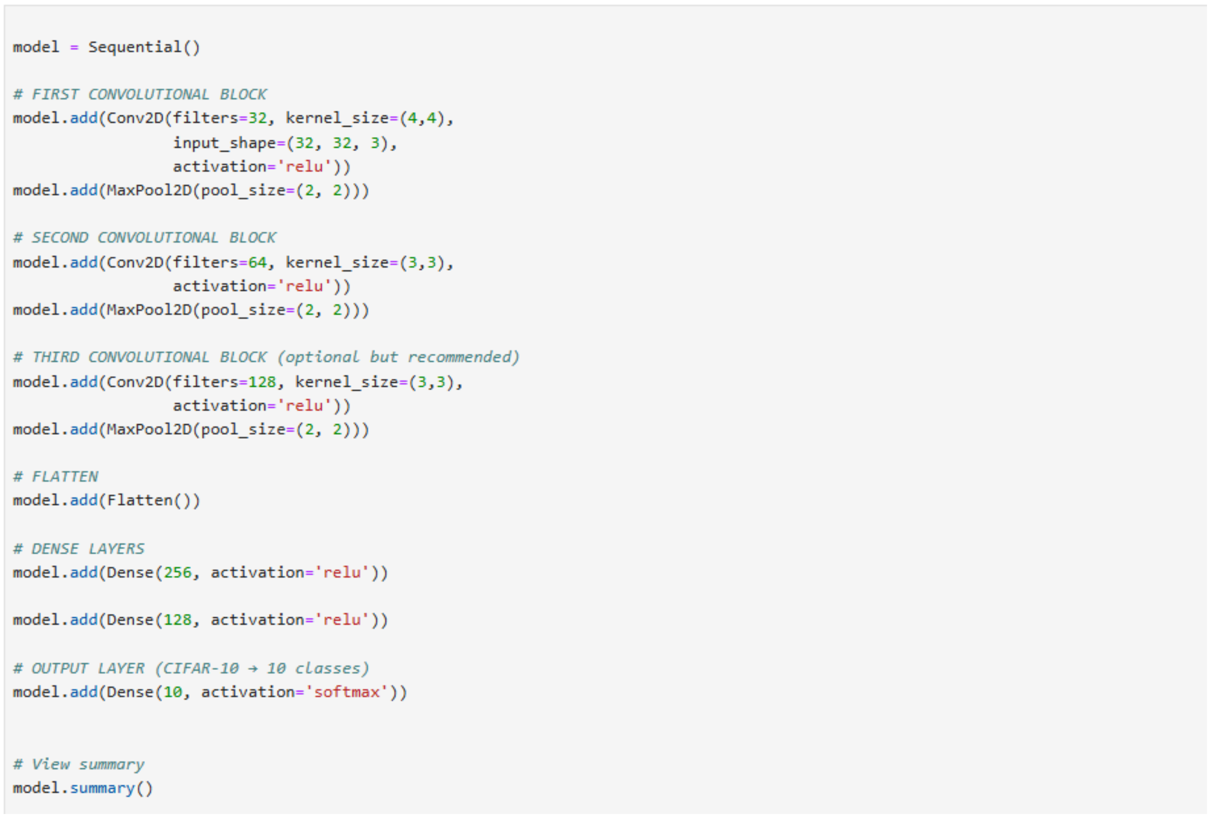
LAB Logbook

Lab 5

Week 5

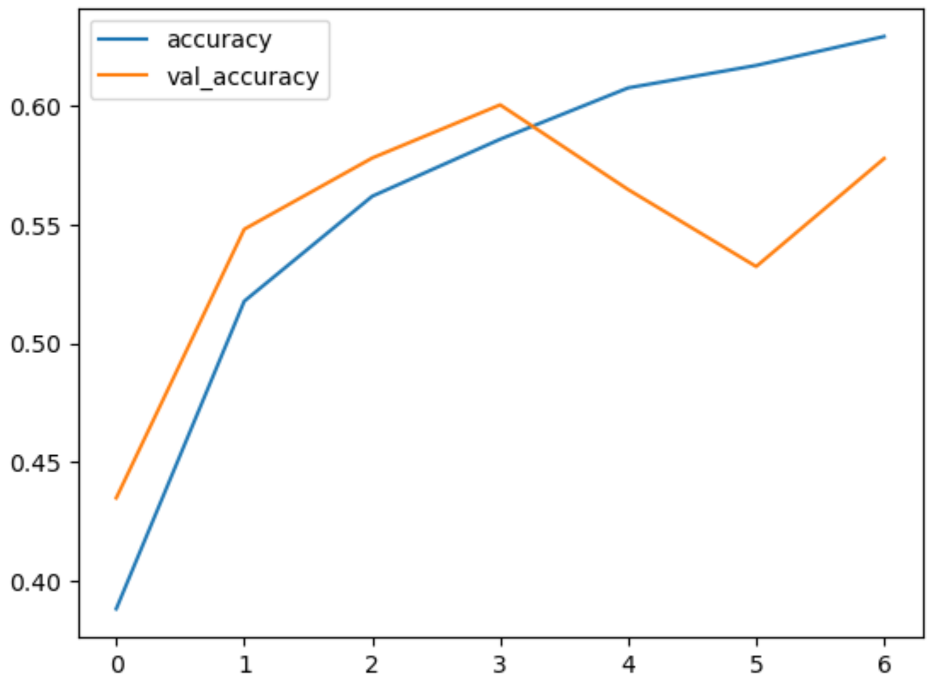


```
model = Sequential()

# FIRST CONVOLUTIONAL BLOCK
model.add(Conv2D(filters=32, kernel_size=(4,4),
                 input_shape=(32, 32, 3),
                 activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

# SECOND CONVOLUTIONAL BLOCK
model.add(Conv2D(filters=64, kernel_size=(3,3),
                 activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

# THIRD CONVOLUTIONAL BLOCK (optional but recommended)
model.add(Conv2D(filters=128, kernel_size=(3,3),
                 activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))

# FLATTEN
model.add(Flatten())

# DENSE LAYERS
model.add(Dense(256, activation='relu'))

model.add(Dense(128, activation='relu'))

# OUTPUT LAYER (CIFAR-10 → 10 classes)
model.add(Dense(10, activation='softmax'))


# View summary
model.summary()
```

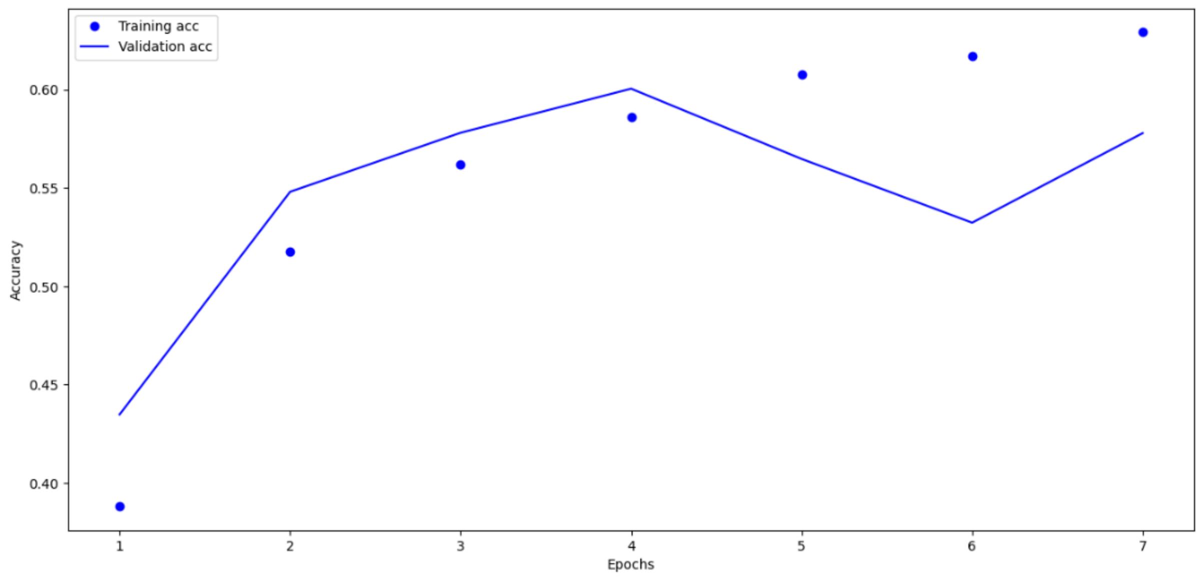
```
Model: "sequential_1"

┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_3 (Conv2D)               │ (None, 29, 29, 32)     │         1,568 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_3 (MaxPooling2D)  │ (None, 14, 14, 32)     │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_4 (Conv2D)               │ (None, 12, 12, 64)     │        18,496 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_4 (MaxPooling2D)  │ (None, 6, 6, 64)       │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ conv2d_5 (Conv2D)               │ (None, 4, 4, 128)      │        73,856 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ max_pooling2d_5 (MaxPooling2D)  │ (None, 2, 2, 128)      │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ flatten_1 (Flatten)             │ (None, 512)            │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_1 (Dense)                 │ (None, 256)            │       131,328 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_2 (Dense)                 │ (None, 128)            │        32,896 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_3 (Dense)                 │ (None, 10)             │         1,290 │
└─────────────────────────────────┴────────────────────────┴───────────────┘

 Total params: 259,434 (1013.41 KB)

 Trainable params: 259,434 (1013.41 KB)

 Non-trainable params: 0 (0.00 B)
```

```
losses[['accuracy', 'val_accuracy']].plot()
```

```
<Axes: >
```

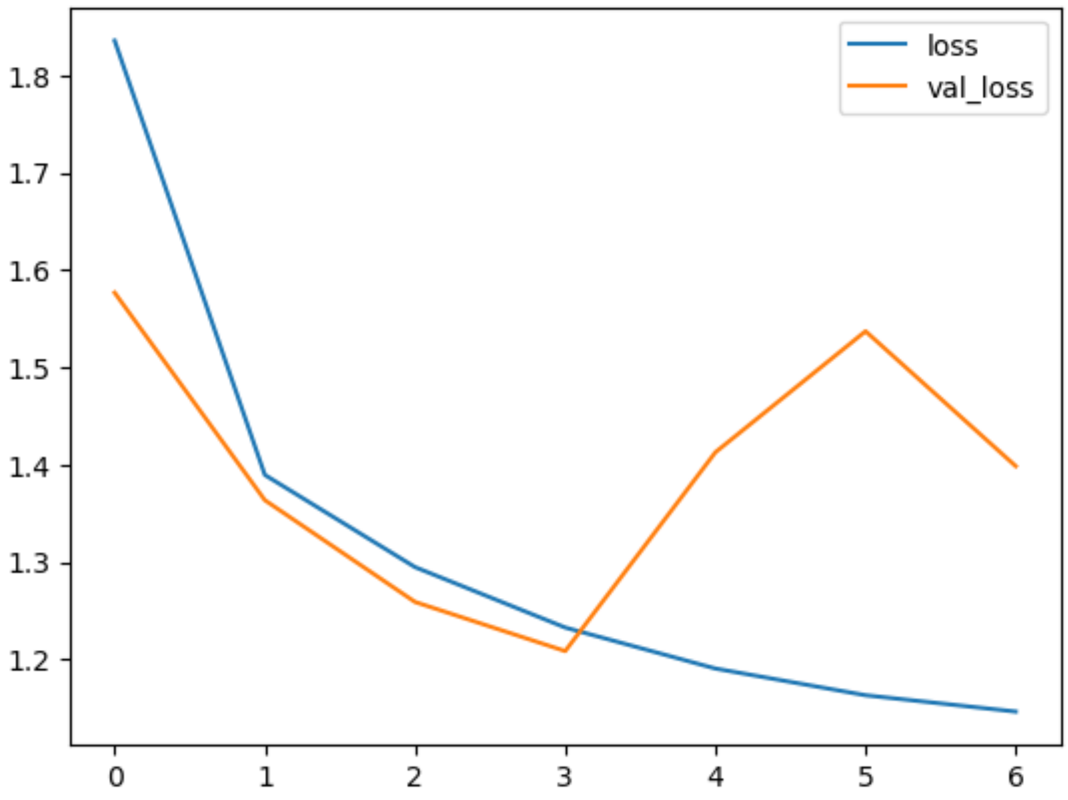


```python
# Plot accuracy and val_accuracy for the neural network training process in more detail

history_dict = history.history

acc_values = history_dict['accuracy']
val_acc_values = history_dict['val_accuracy']
epochs = range(1, len(acc_values) + 1)

plt.figure(num=1, figsize=(15,7))
plt.plot(epochs, acc_values, 'bo', label='Training acc')
plt.plot(epochs, val_acc_values, 'b', label='Validation acc')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
```

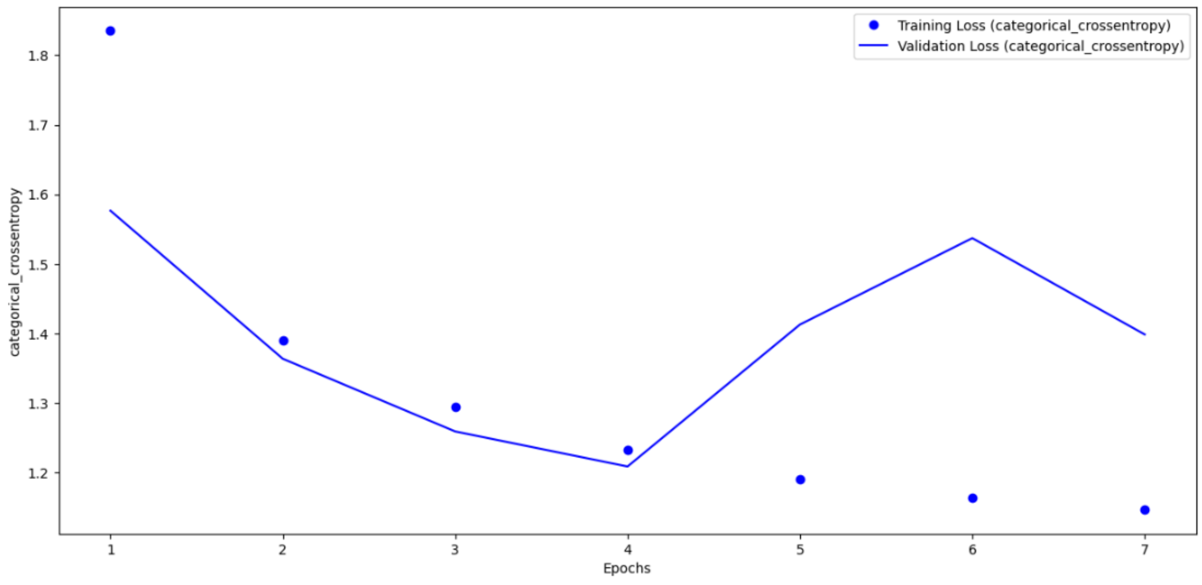


```
losses[['loss', 'val_loss']].plot()
```

```
<Axes: >
```



```
history_dict = history.history
acc_values = history_dict['loss']
val_acc_values = history_dict['val_loss']
epochs = range(1, len(acc_values) +1)
plt.figure(num=1, figsize=(15,7))
plt.plot(epochs, acc_values, 'bo', label='Training Loss (categorical_crossentropy)')
plt.plot(epochs, val_acc_values, 'b', label='Validation Loss (categorical_crossentropy)')
plt.xlabel('Epochs')
plt.ylabel('categorical_crossentropy')
plt.legend()
plt.show()
```


● Training Loss (categorical_crossentropy)
Validation Loss (categorical_crossentropy)
categorical_crossentropy
Epochs