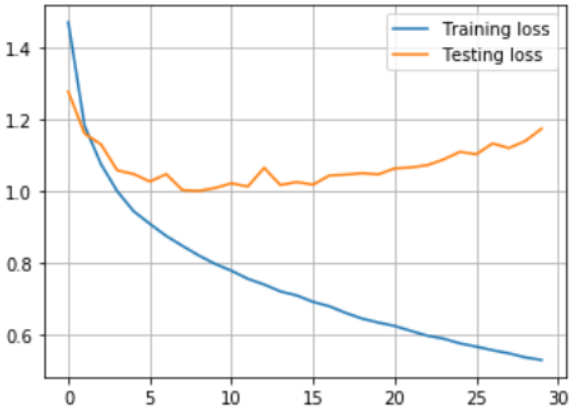
CNN with Keras on CIFAR10 – Experiments Report

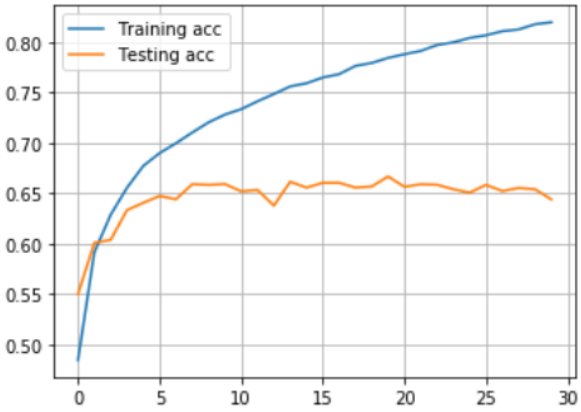
Simple CNN

Hyper-Parameters:

- Epochs = 30
- Batch-Size = 128

CNN	Architecture	Comp. Time s	Acc. Train %	Acc. Test %
1	Layer 1: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 2: **RELU** Layer 3: **MAXPOOL** Pool Size = (2, 2) Layer 4: **FLATTEN** Layer 5: **DENSE** activation = 'softmax'	1060.35	0.8196	0.6440





We can reduce the number of epochs because it looks like we are in the area of overfitting. We set the new number of epochs equal to ten.

Deeper CNN

Hyper- Parameters:

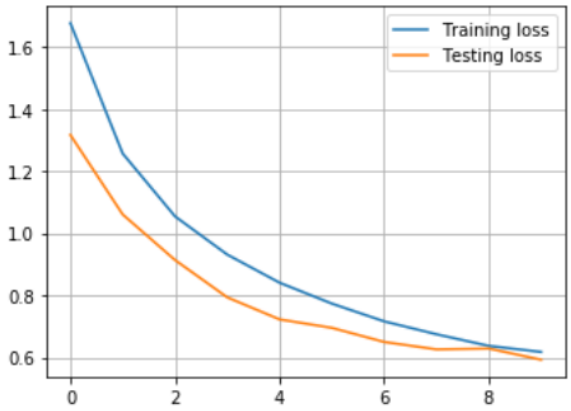
- Epochs = 10
- Batch-Size = 128

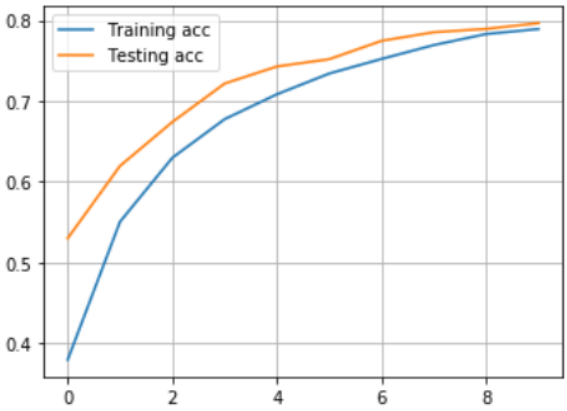
CNN	Architecture	Comp. Time s	Acc. Train %	Acc. Test %
2	Layer 1: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 2: **RELU** Layer 3: **MAXPOOL** Pool Size = (2, 2) Layer 4: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 5: **RELU** Layer 6: **MAXPOOL** Pool Size = (2, 2) Layer 7: **FLATTEN** Layer 8: **DENSE** activation = 'softmax'	800.10	0.7929	0.7265

With the CONV-RELU-POOL architecture, we receive an accuracy of 72.65%. Now we will try the CONV-RELU-CONV-RELU-POOL architecture.

CNN	Architecture	Comp. Time s	Acc. Train %	Acc. Test %
3	Layer 1: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 2: **RELU** Layer 3: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 4: **RELU** Layer 5: **MAXPOOL** Pool Size = (2, 2) Layer 6: **FLATTEN** Layer 7: **DENSE** activation = 'softmax'	2124.51	0.8971	0.6694
4	Layer 1: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 2: **RELU** Layer 3: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 4: **RELU** Layer 5: **MAXPOOL** Pool Size = (2, 2) Layer 6: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 7: **RELU** Layer 8: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 9: **RELU** Layer 10: **MAXPOOL** Pool Size = (2, 2) Layer 11: **FLATTEN** Layer 12: **DENSE** activation = 'softmax'	2773.93	0.8689	0.7474
5	Layer 1: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 2: **RELU** Layer 3: **CONV** D = 32, w = h = 3, S = 1, P = 'same' Layer 4: **RELU** Layer 5: **MAXPOOL** Pool Size = (2, 2) Layer 6: **DROPOUT** rate = 0.25 Layer 7: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 8: **RELU** Layer 9: **CONV** D = 64, w = h = 3, S = 1, P = 'same' Layer 10: **RELU** Layer 11: **MAXPOOL** Pool Size = (2, 2) Layer 12: **DROPOUT** rate = 0.25 Layer 13: **CONV** D = 128, w = h = 3, S = 1, P = 'same' Layer 14: **RELU** Layer 15: **CONV** D = 128, w = h = 3, S = 1, P = 'same' Layer 16: **RELU** Layer 17: **MAXPOOL** Pool Size = (2, 2) Layer 18: **DROPOUT** rate = 0.25 Layer 13: **FLATTEN** Layer 14: **DENSE** activation = 'relu', units = 128 Layer 14: **DROPOUT** rate = 0.5 Layer 14: **DENSE** activation = 'softmax'	3994.04	0.7891	**0.7964**

Now we have a closer look to the 5th model:





It looks like the model hasn't finished learning yet, so we're going through 30 epochs. After 30 epochs we get a test accuracy of **0.8182**.