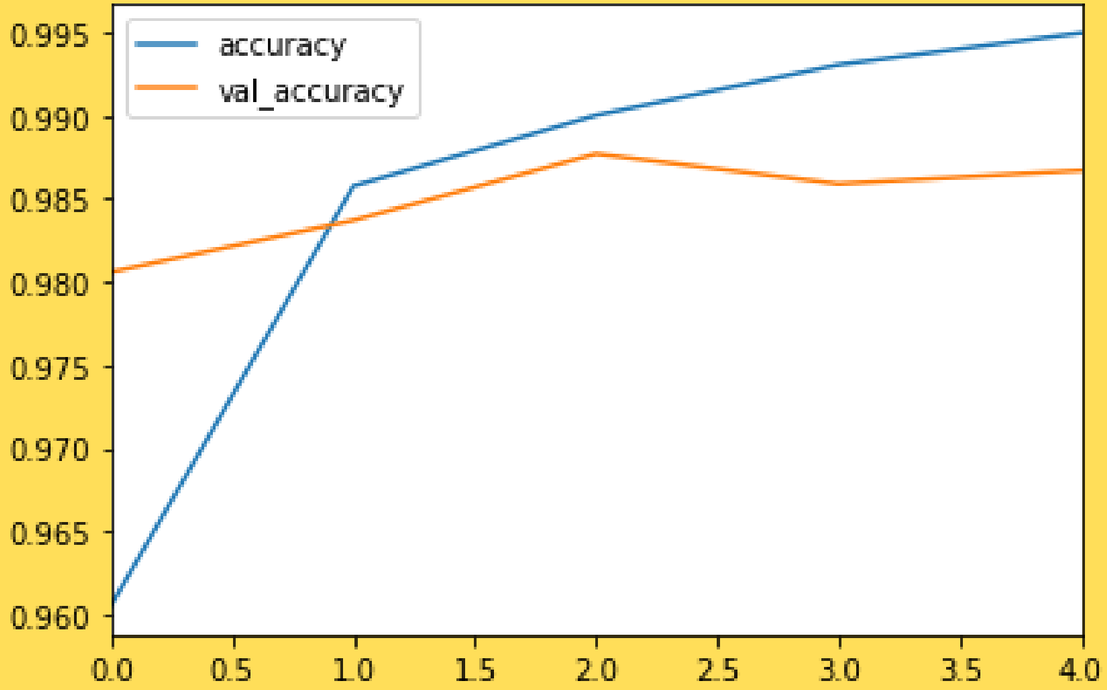
CNN ON MNIST - IO DATASET

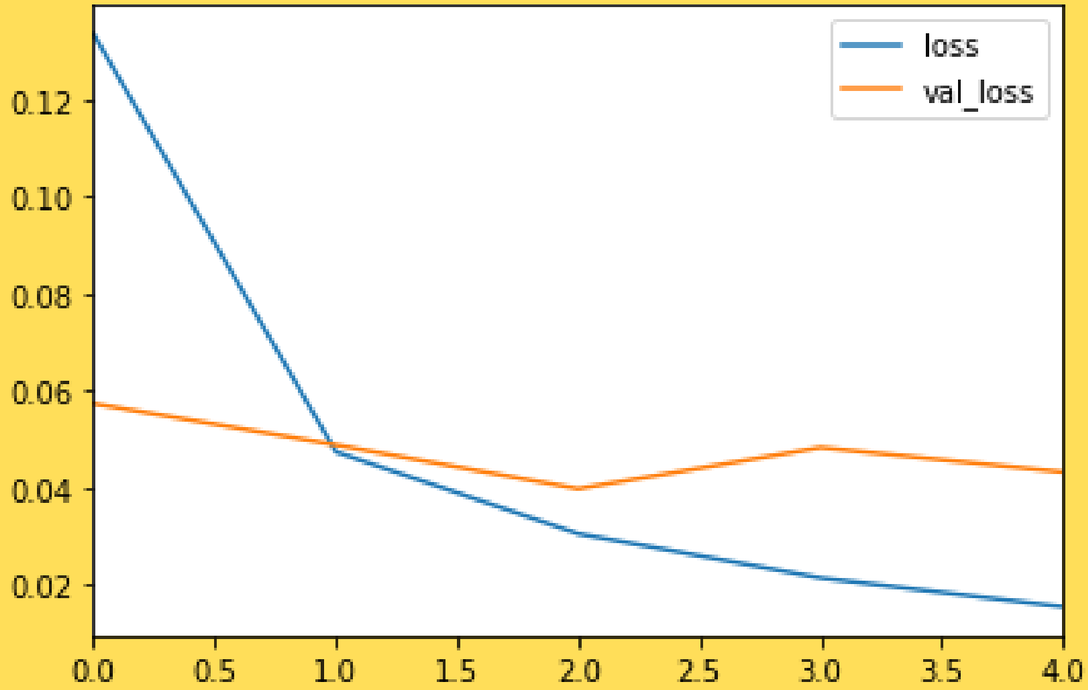
Custom Architecture

99 PERCENT ACCURACY

```
MODEL: "SEQUENTIAL"
_________________________________________________________________
LAYER (TYPE)                 OUTPUT SHAPE              PARAM #
=================================================================
CONV2D (CONV2D)              (NONE, 25, 25, 32)        544
_________________________________________________________________
MAX_POOLING2D (MAXPOOLING2D) (NONE, 12, 12, 32)        0
_________________________________________________________________
FLATTEN (FLATTEN)            (NONE, 4608)              0
_________________________________________________________________
DENSE (DENSE)                (NONE, 128)               589952
_________________________________________________________________
DENSE_1 (DENSE)              (NONE, 10)                1290
=================================================================
TOTAL PARAMS: 591,786
TRAINABLE PARAMS: 591,786
```

```
MODEL = SEQUENTIAL()

# CONVOLUTIONAL LAYER
MODEL.ADD(CONV2D(FILTERS=32, KERNEL_SIZE=
(4,4),INPUT_SHAPE=(28, 28, 1),
ACTIVATION='RELU',))
# POOLING LAYER
MODEL.ADD(MAXPOOL2D(POOL_SIZE=(2, 2)))

# FLATTEN IMAGES FROM 28 BY 28 TO 764 BEFORE
FINAL LAYER
MODEL.ADD(FLATTEN())

# 128 NEURONS IN DENSE HIDDEN LAYER (YOU CAN
CHANGE THIS NUMBER OF NEURONS)
MODEL.ADD(DENSE(128, ACTIVATION='RELU'))

# LAST LAYER IS THE CLASSIFIER, THUS 10 POSSIBLE
CLASSES
MODEL.ADD(DENSE(10, ACTIVATION='SOFTMAX'))

# HTTPS://KERAS.IO/METRICS/
MODEL.COMPILE(LOSS='CATEGORICAL_CROSSENTROPY'
,
        OPTIMIZER='ADAM',
        METRICS=['ACCURACY']) # WE CAN ADD IN
ADDITIONAL METRICS HTTPS://KERAS.IO/METRICS/
```





	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.99	0.99	0.99	980
1	0.99	1.00	0.99	1135
2	0.99	0.98	0.98	1032
3	0.99	0.99	0.99	1010
4	0.97	1.00	0.98	982
5	0.99	0.99	0.99	892
6	1.00	0.98	0.99	958
7	0.97	1.00	0.98	1028
8	0.99	0.98	0.98	974
9	0.99	0.96	0.98	1009
ACCURACY			0.99	10000
MACRO AVG	0.99	0.99	0.99	10000
WEIGHTED AVG	0.99	0.99	0.99	10000

```
ARRAY([[ 975,    0,    1,    0,    0,    0,    3,    1,    0,    0],
       [   0, 1134,    0,    0,    0,    0,    0,    1,    0,    0],
       [   1,    4, 1007,    1,    4,    0,    0,   13,    2,    0],
       [   0,    0,    5, 1000,    0,    2,    0,    2,    1,    0],
       [   0,    0,    0,    0,  979,    0,    0,    0,    0,    3],
       [   0,    0,    0,    6,    0,  883,    1,    0,    2,    0],
       [   4,    3,    0,    1,    5,    1,  943,    0,    1,    0],
       [   0,    1,    2,    0,    0,    0,    0, 1024,    1,    0],
       [   2,    1,    4,    4,    4,    0,    0,    5,  952,    2],
       [   0,    3,    0,    1,   16,    4,    0,   14,    1,  970]],
                              DTYPE=INT64)
```