https://colab.research.google.com/drive/1_xPlkxt8HGHeamcKr8FMPkCkkqOLPW0w#scrollTo=3NS_ZNnQ2Q3u

New Tab | Home | Small_MNIST_Convnet | Hello, Colaboratory - Colabora | Simple_MNIST_CNN - Colabor

https://colab.research.google.com/drive/1_xPlkxt8HGHeamcKr8FMPkCkkqOLPW0w#scrollTo=3NS_ZNnQ2Q3u

```
[20] 54000/54000 [==============================] - 5s 94us/step - loss: 0.0139 - acc: 0.9956 - val_loss: 0.0344 - val_acc: 0.9922
     Epoch 10/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0114 - acc: 0.9964 - val_loss: 0.0389 - val_acc: 0.9912
     Epoch 11/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0112 - acc: 0.9963 - val_loss: 0.0359 - val_acc: 0.9905
     Epoch 12/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0088 - acc: 0.9969 - val_loss: 0.0301 - val_acc: 0.9927
     Epoch 13/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0093 - acc: 0.9966 - val_loss: 0.0342 - val_acc: 0.9915
     Epoch 14/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0078 - acc: 0.9973 - val_loss: 0.0323 - val_acc: 0.9912
     Epoch 15/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0061 - acc: 0.9979 - val_loss: 0.0375 - val_acc: 0.9920
     Epoch 16/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0072 - acc: 0.9976 - val_loss: 0.0379 - val_acc: 0.9923
     Epoch 17/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0059 - acc: 0.9980 - val_loss: 0.0362 - val_acc: 0.9907
     Epoch 18/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0057 - acc: 0.9979 - val_loss: 0.0456 - val_acc: 0.9915
     Epoch 19/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0044 - acc: 0.9984 - val_loss: 0.0370 - val_acc: 0.9913
     Epoch 20/20
     54000/54000 [==============================] - 5s 94us/step - loss: 0.0047 - acc: 0.9984 - val_loss: 0.0460 - val_acc: 0.9920
```

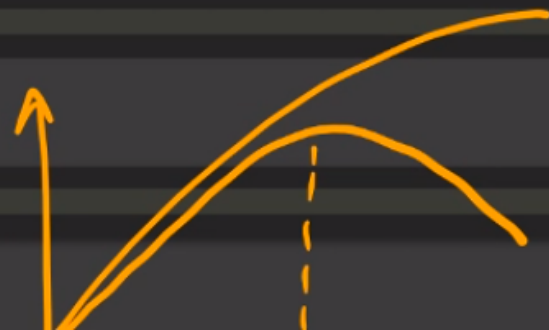
valsplit

Best hyperparam

```
[21] 1 model.evaluate(XTest,YTest)
```

```
10000/10000 [==============================] - 1s 87us/step
[0.034718466649535186, 0.9914]
```

12 epochs



```
1
```

New Tab | Home | Small_MNIST_Convnet | Hello, Colaboratory - Colabora | Simple_MNIST_CNN - Colabor

https://colab.research.google.com/drive/1_xPlkxt8HGHeamcKr8FMPkCkkqOLPW0w#scrollTo=3NS_ZNnQ2Q3u

[19]
```
def preprocess_data(X,Y):
    X = X.reshape((-1,28,28,1))
    X = X/255.0
    Y = to_categorical(Y)
    return X,Y

XTrain,YTrain = preprocess_data(XTrain,YTrain)
print(XTrain.shape,YTrain.shape)

XTest,YTest = preprocess_data(XTest,YTest)
print(XTest.shape,YTest.shape)
```

```
(60000, 28, 28, 1) (60000, 10)
(10000, 28, 28, 1) (10000, 10)
```


Complete Data ↓ model.fit(XTrain, YTrain)

[20]
```
# Model Compilation
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
hist = model.fit(XTrain,YTrain,epochs=20,validation_split=0.1,batch_size=128)
```

```
Train on 54000 samples, validate on 6000 samples
Epoch 1/20
54000/54000 [==============================] - 6s 108us/step - loss: 0.2652 - acc: 0.9238 - val_loss: 0.0595 - val_acc: 0.9840
Epoch 2/20
54000/54000 [==============================] - 5s 96us/step - loss: 0.0675 - acc: 0.9791 - val_loss: 0.0488 - val_acc: 0.9853
Epoch 3/20
54000/54000 [==============================] - 5s 95us/step - loss: 0.0448 - acc: 0.9862 - val_loss: 0.0339 - val_acc: 0.9910
Epoch 4/20
54000/54000 [==============================] - 5s 95us/step - loss: 0.0346 - acc: 0.9892 - val_loss: 0.0390 - val_acc: 0.9890
Epoch 5/20
54000/54000 [==============================] - 5s 94us/step - loss: 0.0281 - acc: 0.9910 - val_loss: 0.0421 - val_acc: 0.9883
Epoch 6/20
54000/54000 [==============================] - 5s 94us/step - loss: 0.0232 - acc: 0.9927 - val_loss: 0.0300 - val_acc: 0.9910
Epoch 7/20
54000/54000 [==============================] - 5s 94us/step - loss: 0.0194 - acc: 0.9938 - val_loss: 0.0458 - val_acc: 0.9858
Epoch 8/20
54000/54000 [==============================] - 5s 94us/step - loss: 0.0171 - acc: 0.9944 - val_loss: 0.0377 - val_acc: 0.9905
```

New Tab | Home | Small_MNIST_Convnet | Hello, Colaboratory - Colabora | Simple_MNIST_CNN - Colabor

https://colab.research.google.com/drive/1_xPlkxt8HGHeamcKr8FMPkCkkqOLPW0w#scrollTo=3NS_ZNnQ2Q3u

Simple_MNIST_CNN

File Edit View Insert Runtime Tools Help

CODE TEXT CELL CELL — COMMENT — RAM Disk

[2]
```python
from keras.layers import *
from keras.models import Sequential
```

```
Using TensorFlow backend.
```

[17]
```python
# Build a Model

model = Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
model.summary()
```

rohit5840@gmail.com

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_5 (MaxPooling2	(None, 13, 13, 32)	0
conv2d_8 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_6 (MaxPooling2	(None, 5, 5, 64)	0

Larger image.

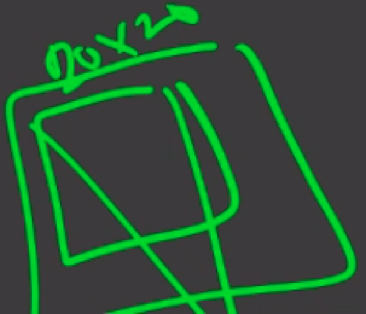
3x3 filters

5x5

more layers

conv pool

Pooling → Downscale H,W very quickly → Larger Receptive Field



Simple_MNIST_CNN

```
from keras.layers import *
from keras.models import Sequential
```

```
Using TensorFlow backend.
```

```
# Build a Model

model = Sequential()
model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(MaxPool2D((2,2)))
model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
model.add(Flatten())
model.add(Dense(64,activation='relu'))
model.add(Dense(10,activation='softmax'))
model.summary()
```

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_5 (MaxPooling2	(None, 13, 13, 32)	0
conv2d_8 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_6 (MaxPooling2	(None, 5, 5, 64)	0

New Tab | Home | Small_MNIST_Convnet | Hello, Colaboratory - Colabora | Simple_MNIST_CNN - Colabor

https://colab.research.google.com/drive/1_xPlkxt8HGHeamcKr8FMPkCkkqOLPW0w#scrollTo=WST-1Hu11PYs

```
[2] 2 from keras.models import Sequential
```

```
Using TensorFlow backend.
```

```
1  # Build a Model
2
3  model = Sequential()
4  model.add(Conv2D(32,(3,3),activation='relu',input_shape=(28,28,1)))
5  model.add(MaxPooling2D((2,2)))
6  model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
7  model.add(MaxPooling2D((2,2)))
8  model.add(Conv2D(64,(3,3),activation='relu',input_shape=(28,28,1)))
9  model.add(Flatten())
10 model.add(Dense(64,activation='relu'))
11 model.add(Dense(10,activation='softmax'))
12 model.summary()
```


64 → 20 output

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 3, 3, 64)          36928
_________________________________________________________________
flatten_3 (Flatten)          (None, 576)               0
_________________________________________________________________
dense_5 (Dense)              (None, 64)                36928
_________________________________________________________________
dense_6 (Dense)              (None, 10)                650
=================================================================
Total params: 93,322
```

→ increase the no channel

→ 3x3 filters

→ Pooling (2,2) (2,2)

→