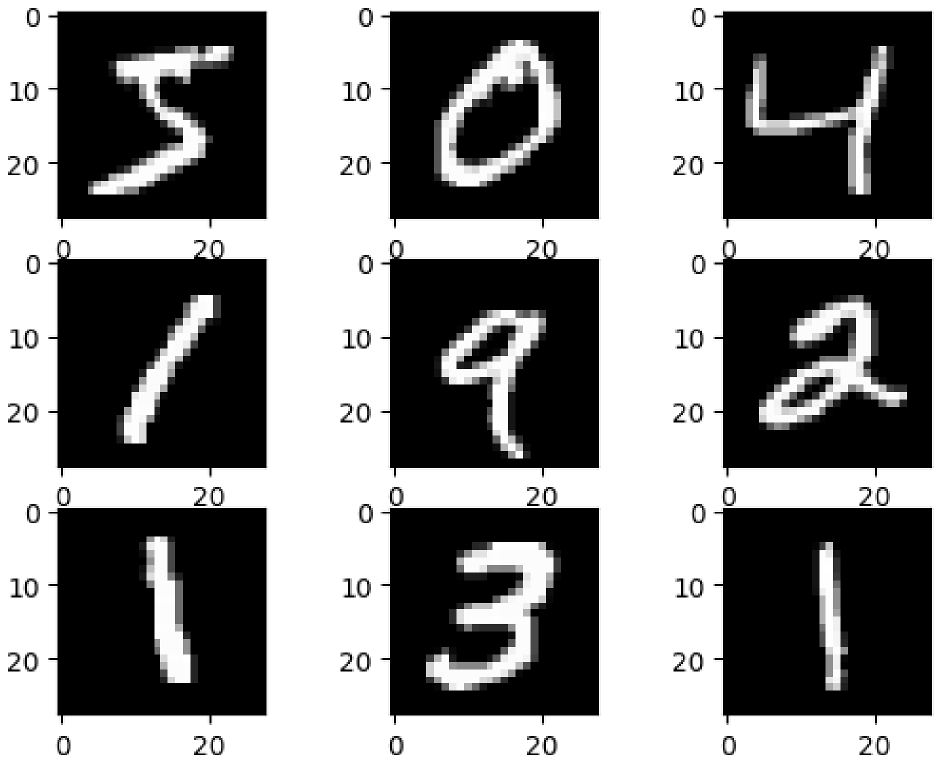
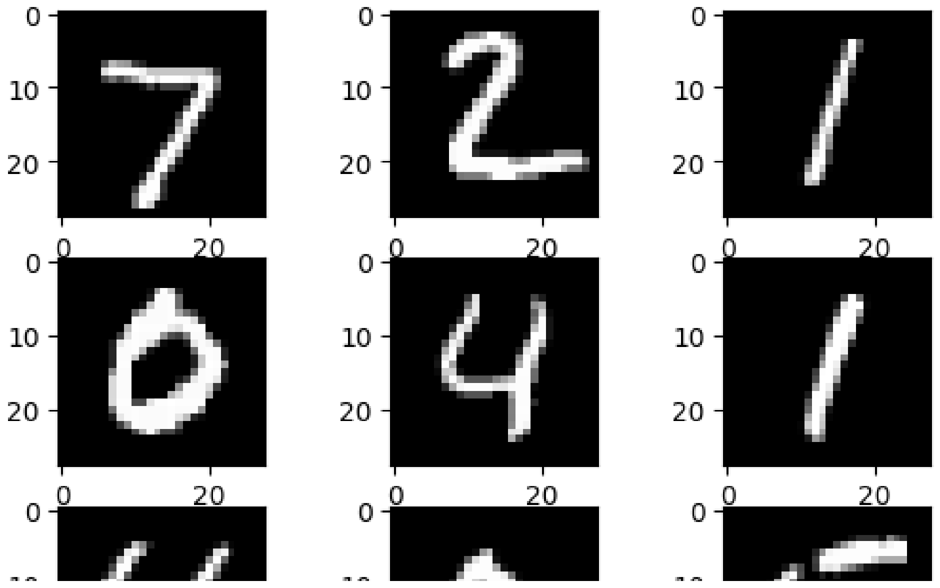
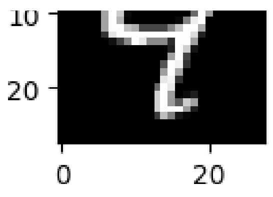
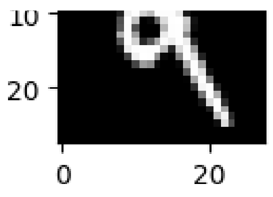
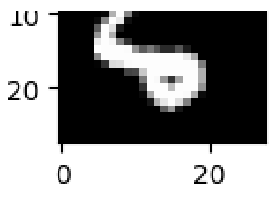
```
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt

(trainX, trainy), (testX, testy) = mnist.load_data()
print('Train: X=%s, y=%s' % (trainX.shape, trainy.shape))
print('Test: X=%s, y=%s' % (testX.shape, testy.shape))

for i in range(9):
  plt.subplot(330 + 1 + i)
  plt.imshow(trainX[i], cmap=plt.get_cmap('gray'))
  print(trainy[i])

plt.show()
plt.clf()

for i in range(9):
  plt.subplot(330 + 1 + i)
  plt.imshow(testX[i], cmap=plt.get_cmap('gray'))
  print(testy[i])

plt.show()
```

```
Train: X=(60000, 28, 28), y=(60000,)
Test: X=(10000, 28, 28), y=(10000,)
5
0
4
1
9
2
1
3
1
```



```
7
2
1
0
4
1
4
9
5
```




10
20
0
20

10
20
0
20

10
20
0
20

```
from numpy import mean
from numpy import std
from matplotlib import pyplot as plt
from sklearn.model_selection import KFold
from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.utils import plot_model
from google.colab import files
from sklearn.model_selection import train_test_split

DEBUGVIS = True


(trainX, trainY), (testX, testY) = mnist.load_data()
trainX = trainX.reshape((trainX.shape[0], 28, 28, 1))
testX = testX.reshape((testX.shape[0], 28, 28, 1))
trainY = to_categorical(trainY)
testY = to_categorical(testY)

if DEBUGVIS:
  print(testY[1])
  plt.subplot(330 + 1)
  plt.imshow(testX[1], cmap=plt.get_cmap('gray'))#
  plt.show()

trainX = trainX.astype('float32')
testX = testX.astype('float32')
trainX = trainX / 255.0
testX = testX / 255.0

model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', \
                   kernel_initializer='he_uniform', \
                   input_shape=(28, 28, 1)))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(100, activation='relu', \
                  kernel_initializer='he_uniform'))
model.add(Dense(10, activation='softmax'))
opt = SGD(learning_rate=0.01, momentum=0.9)
model.compile(optimizer=opt, loss='categorical_crossentropy', \
                metrics=['accuracy'])

if DEBUGVIS:
  filename = 'keras_model_plot.png'
  plot_model(model, \
           to_file=filename,
           show_shapes=True,
           show_layer_names=True)
  %ls
  files.download(filename)
```

```
trainX, validX, trainY, validY = train_test_split(trainX, trainY, \
                                                  test_size=0.20, random_state=42

history = model.fit(trainX, trainY, epochs=10, \
            batch_size=32, validation_data=(validX, validY), verbose=1)
_, acc = model.evaluate(validX, validY, verbose=0)
print('Validation Set > %.3f' % (acc * 100.0))
_, acc = model.evaluate(trainX, trainY, verbose=0)
print('Training Set > %.3f' % (acc * 100.0))

plt.clf()
plt.title('Classification Accuracy')
plt.plot(history.history['accuracy'], color='blue', label='train')
```