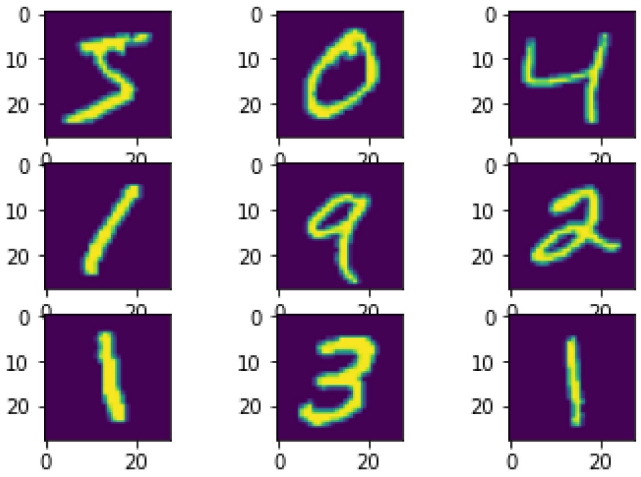
Trương Duy Kha 19146015

https://github.com/TruongDuyKha/Mnist.git

```
from keras.datasets import mnist
import matplotlib.pyplot as plt
from tensorflow.keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Activation, Dropout, Dense
from tensorflow.keras.optimizers import RMSprop
from keras.models import load_model
import numpy as np
```

```
(x_train, y_train),(x_test,y_test) = mnist.load_data()
```

```
for i in range(9):
  plt.subplot(330 + i +1)
  plt.imshow(x_train[i])
plt.show()
```



```
x = x_test
x_train = x_train.reshape(60000,784)
x_test = x_test.reshape(10000,784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test/= 255
```

```
y_train = to_categorical(y_train,10)
y_test = to_categorical(y_test,10)
```

```
model = Sequential()
model.add(Dense(512,activation = 'relu',input_shape = (784,)))
model.add(Dropout(0.2))
model.add(Dense(512,activation = 'relu'))
model.add(Dropout(0.2))
```

```
model.add(Dense(10,activation='softmax'))
model.summary()
model.compile(loss = 'categorical_crossentropy', optimizer = 'RMSprop', metrics = ['accuracy
history = model.fit(x_train,y_train,batch_size= 128,epochs = 10, verbose = 1, validation_dat
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 512)               401920

 dropout_4 (Dropout)         (None, 512)               0

 dense_7 (Dense)             (None, 512)               262656

 dropout_5 (Dropout)         (None, 512)               0

 dense_8 (Dense)             (None, 10)                5130

=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
_________________________________________________________________
Epoch 1/10
469/469 [==============================] - 10s 21ms/step - loss: 0.2455 - accuracy: 0.9
Epoch 2/10
469/469 [==============================] - 9s 20ms/step - loss: 0.1015 - accuracy: 0.96
Epoch 3/10
469/469 [==============================] - 9s 19ms/step - loss: 0.0752 - accuracy: 0.97
Epoch 4/10
469/469 [==============================] - 9s 19ms/step - loss: 0.0582 - accuracy: 0.98
Epoch 5/10
469/469 [==============================] - 9s 19ms/step - loss: 0.0500 - accuracy: 0.98
Epoch 6/10
469/469 [==============================] - 9s 19ms/step - loss: 0.0429 - accuracy: 0.98
Epoch 7/10
469/469 [==============================] - 10s 21ms/step - loss: 0.0384 - accuracy: 0.9
Epoch 8/10
469/469 [==============================] - 9s 19ms/step - loss: 0.0330 - accuracy: 0.99
Epoch 9/10
469/469 [==============================] - 12s 25ms/step - loss: 0.0308 - accuracy: 0.9
Epoch 10/10
469/469 [==============================] - 10s 22ms/step - loss: 0.0289 - accuracy: 0.9
```

```
score = model.evaluate(x_test,y_test,verbose = 1)
model.save('Mnist_ANN.h5')
```

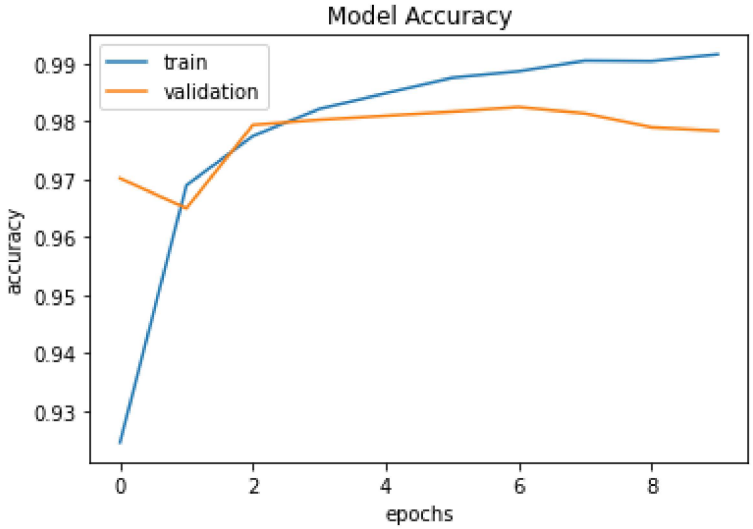
```
313/313 [==============================] - 1s 3ms/step - loss: 0.1201 - accuracy: 0.978
```

```
Model = load_model('Mnist_ANN.h5')
```

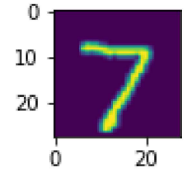
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
```

```
plt.ylabel('accuracy')
plt.xlabel('epochs')
plt.legend(['train','validation'],loc='upper_left')
```

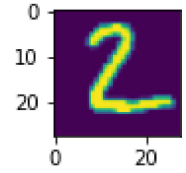
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: MatplotlibDeprecationWa
        best
        upper right
        upper left
        lower left
        lower right
        right
        center left
        center right
        lower center
        upper center
        center
This will raise an exception in 3.3.

<matplotlib.legend.Legend at 0x7f59b3012e10>
```



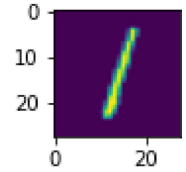

```
y_pred = Model.predict(x_test)
for i in range(9):
  plt.subplot(330 + i + 1)
  plt.imshow(x[i])
  plt.show()
  print(np.argmax(y_pred[0:10],axis = 1)[i])
```


0
10
20
0
20

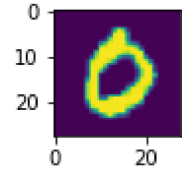
7


0
10
20
0
20

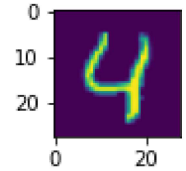
2


0
10
20
0
20

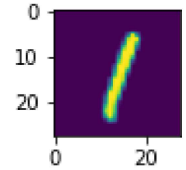
1


0
10
20
0
20

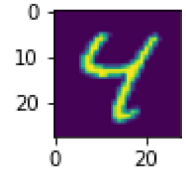
0


0
10
20
0
20

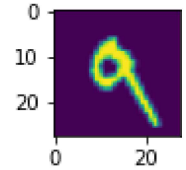
4


0
10
20
0
20

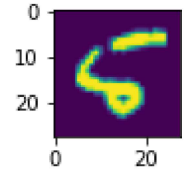
1


0
10
20
0
20

4


0
10
20
0
20

9


0
10
20
0
20

5