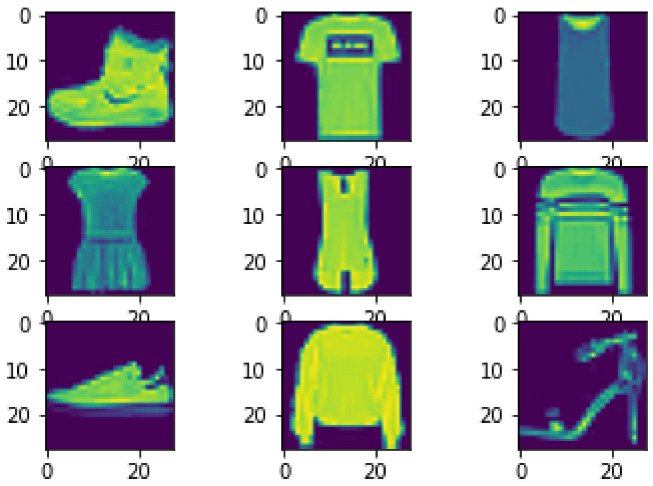
```
#Trần Nam phương
```

```
from keras.datasets import fashion_mnist
import matplotlib.pyplot as plt
(x_train,y_train),(x_test,y_test)=fashion_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_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
```

```
from tensorflow.keras.utils import to_categorical
```

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

```
    (60000, 10)
```

```
from keras.models import Sequential
from keras.layers import Dense,Activation,Dropout
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(0.1))
model.add(Dense(10,activation="softmax"))
model.summary()
```

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

      dropout (Dropout)           (None, 512)               0

      dense_1 (Dense)             (None, 512)               262656

      dropout_1 (Dropout)         (None, 512)               0

      dense_2 (Dense)             (None, 10)                5130

     =================================================================
     Total params: 669,706
     Trainable params: 669,706
     Non-trainable params: 0
     _________________________________________________________________
```

```
from tensorflow.keras.optimizers import RMSprop
model.compile(loss='categorical_crossentropy',optimizer=RMSprop(),metrics=['accuracy'])
```

```
model.fit(x_train, y_train, batch_size=128,epochs =20, verbose=1,validation_data=(x_test,y_te
```

```
     Epoch 1/20
     469/469 [==============================] - 11s 21ms/step - loss: 0.5583 - accuracy: 0.79
     Epoch 2/20
     469/469 [==============================] - 10s 21ms/step - loss: 0.4007 - accuracy: 0.85
     Epoch 3/20
     469/469 [==============================] - 9s 20ms/step - loss: 0.3608 - accuracy: 0.867
     Epoch 4/20
     469/469 [==============================] - 10s 21ms/step - loss: 0.3443 - accuracy: 0.87
     Epoch 5/20
     469/469 [==============================] - 10s 22ms/step - loss: 0.3315 - accuracy: 0.87
     Epoch 6/20
     469/469 [==============================] - 9s 20ms/step - loss: 0.3189 - accuracy: 0.884
     Epoch 7/20
     469/469 [==============================] - 10s 20ms/step - loss: 0.3098 - accuracy: 0.88
     Epoch 8/20
     469/469 [==============================] - 10s 22ms/step - loss: 0.3030 - accuracy: 0.89
     Epoch 9/20
     469/469 [==============================] - 11s 23ms/step - loss: 0.2976 - accuracy: 0.89
     Epoch 10/20
     469/469 [==============================] - 9s 20ms/step - loss: 0.2936 - accuracy: 0.894
     Epoch 11/20
     469/469 [==============================] - 10s 20ms/step - loss: 0.2874 - accuracy: 0.89
     Epoch 12/20
```

```
469/469 [==============================] - 9s 20ms/step - loss: 0.2814 - accuracy: 0.899
Epoch 13/20
469/469 [==============================] - 10s 20ms/step - loss: 0.2770 - accuracy: 0.90
Epoch 14/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2700 - accuracy: 0.90
Epoch 15/20
469/469 [==============================] - 11s 23ms/step - loss: 0.2730 - accuracy: 0.90
Epoch 16/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2679 - accuracy: 0.90
Epoch 17/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2653 - accuracy: 0.90
Epoch 18/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2602 - accuracy: 0.90
Epoch 19/20
469/469 [==============================] - 10s 20ms/step - loss: 0.2576 - accuracy: 0.90
Epoch 20/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2580 - accuracy: 0.90
<keras.callbacks.History at 0x7fcee5a58310>
```

```
model.save('ANNMNIST.h5')
```

```
history=model.fit(x_train, y_train, batch_size=128,epochs =20, verbose=1,validation_data=(x_t
```

```
Epoch 1/20
469/469 [==============================] - 10s 22ms/step - loss: 0.2500 - accuracy: 0.91
Epoch 2/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2498 - accuracy: 0.91
Epoch 3/20
469/469 [==============================] - 10s 22ms/step - loss: 0.2493 - accuracy: 0.91
Epoch 4/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2467 - accuracy: 0.91
Epoch 5/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2441 - accuracy: 0.91
Epoch 6/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2435 - accuracy: 0.91
Epoch 7/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2408 - accuracy: 0.91
Epoch 8/20
469/469 [==============================] - 11s 23ms/step - loss: 0.2408 - accuracy: 0.91
Epoch 9/20
469/469 [==============================] - 10s 22ms/step - loss: 0.2362 - accuracy: 0.91
Epoch 10/20
469/469 [==============================] - 9s 20ms/step - loss: 0.2315 - accuracy: 0.918
Epoch 11/20
469/469 [==============================] - 10s 20ms/step - loss: 0.2358 - accuracy: 0.91
Epoch 12/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2285 - accuracy: 0.91
Epoch 13/20
469/469 [==============================] - 10s 22ms/step - loss: 0.2325 - accuracy: 0.91
Epoch 14/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2291 - accuracy: 0.92
Epoch 15/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2257 - accuracy: 0.92
```

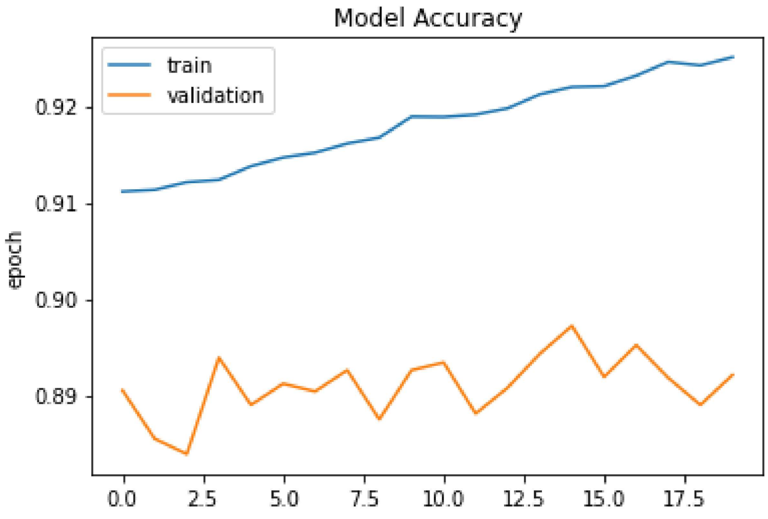
```
Epoch 16/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2291 - accuracy: 0.92
Epoch 17/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2218 - accuracy: 0.92
Epoch 18/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2203 - accuracy: 0.92
Epoch 19/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2164 - accuracy: 0.92
Epoch 20/20
469/469 [==============================] - 10s 21ms/step - loss: 0.2200 - accuracy: 0.92
```

```
score = model.evaluate(x_test, y_test, verbose=1)
print('Test loss =', score[0])
print('Test accurracy =', score[1])
```

```
313/313 [==============================] - 2s 6ms/step - loss: 0.5625 - accuracy: 0.892
Test loss = 0.5625239014625549
Test accurracy = 0.8920999765396118
```

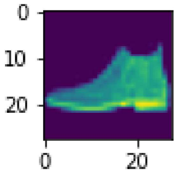
```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('accuracy')
plt.ylabel('epoch')
plt.legend(['train','validation'],loc='upper left')
```

```
<matplotlib.legend.Legend at 0x7fcee216d910>
```

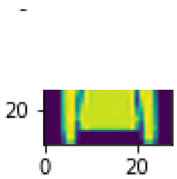


```
from keras.datasets import fashion_mnist
y_pred=model.predict(x_test)
for i in range(9):
  plt.subplot(330+i+1)
  plt.imshow(x[i])
  print(y_pred[i])
  plt.show()
```

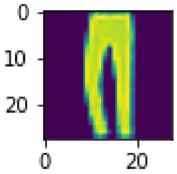
```
[5.4563914e-17 4.4429199e-19 5.1293770e-20 5.7831017e-22 4.3914724e-21
 1.3081070e-08 1.7146519e-17 1.3792871e-07 2.4911008e-20 9.9999988e-01]
```



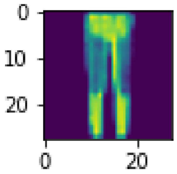
```
[3.1566785e-08 4.6189273e-30 9.9994254e-01 5.0363153e-14 2.8488521e-05
 4.2667805e-36 2.9016681e-05 0.0000000e+00 1.9389879e-23 0.0000000e+00]
```



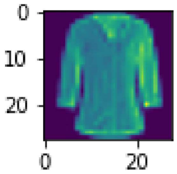
```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```



```
[0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
```



```
[1.2887251e-03 2.1201867e-23 1.2057430e-04 1.1664804e-06 1.0013320e-05
 9.2782493e-17 9.9857950e-01 1.7744533e-21 4.4715392e-13 4.6321416e-19]
```



✓ 4 giây hoàn thành lúc 16:40