



# DEEP LEARNING KERAS

Phạm Nguyên Khang  
[pnkhang@cit.ctu.edu.vn](mailto:pnkhang@cit.ctu.edu.vn)

5/2020



CANTHO UNIVERSITY

# Giới thiệu

- Keras
  - API mức cao dùng để xây dựng và huấn luyện các mô hình học sâu.
  - Xây dựng nhanh
  - Thân thiện, dễ sử dụng
  - Mô-đun, dễ kết hợp
  - Dễ mở rộng



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Huấn luyện:
  - 6000 ảnh 28 x 28
- Kiểm thử:
  - 1000 ảnh 28 x 28
- Số lớp:
  - 10, đánh số từ 0 - 9





CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Import

```
import tensorflow as tf  
from tensorflow import keras  
import numpy as np  
import matplotlib.pyplot as plt
```

- Download tập dữ liệu:

```
fashion_mnist = keras.datasets.fashion_mnist
```

```
(train_images, train_labels), (test_images, test_labels) =  
fashion_mnist.load_data()
```

- Tên lớp:

```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress',  
'Coat', 'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Khám phá tập dữ liệu:

`train_images.shape` (6000, 28, 28)

`len(train_labels)` 6000

`train_labels`

`array([9, 0, 0, ..., 3, 0, 5], dtype=uint8)`

- Xem ảnh:

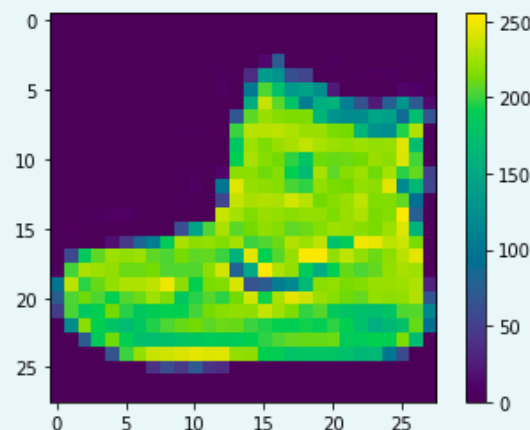
`plt.figure()`

`plt.imshow(train_images[0])`

`plt.colorbar()`

`plt.grid(False)`

`plt.show()`





CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Tiền xử lý:
  - Đưa giá trị mức xám về khoảng  $[0, 1]$ .  
 $\text{train\_images} = \text{train\_images} / 255.0$   
 $\text{test\_images} = \text{test\_images} / 255.0$



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Xây dựng mô hình
  - Mạng nơ-ron 3 tầng
    - Tầng làm phẳng (Flatten): ma trận  $\rightarrow$  vector.
    - Tầng kết nối đầy đủ (Dense), hàm kích hoạt RELU
    - Tầng đầu ra kết nối đầy đủ (Dense), không có hàm kích hoạt.
  - Code:

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28, 28)),
    keras.layers.Dense(128, activation='relu'),
    keras.layers.Dense(10)
])
```



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Biên dịch mô hình (compile)
  - Gắn kết mô hình với bộ tối ưu, hàm loss, và tiêu chí đánh giá trong quá trình huấn luyện
    - Optimizer: adam (một dạng của Gradient descent)
    - Loss: SparseCategoricalCrossentropy
    - Metrics: accuracy
  - Code:

```
model.compile(optimizer='adam',  
loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True), metrics=['accuracy'])
```





CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Huấn luyện mô hình (train)
  - Đưa dữ liệu huấn luyện gồm ảnh và nhãn vào mô hình
    - train\_images, train\_labels
  - Mô hình sẽ học cách liên kết ảnh => nhãn
  - Dự đoán nhãn của ảnh trong tập test
    - test\_images
  - Kiểm tra kết quả dự đoán với nhãn thật
    - test\_labels
- Code:  
`model.fit(train_images, train_labels, epochs=10)`



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

Train on 60000 samples

Epoch 1/10

60000/60000 [=====] - 4s 61us/sample - loss: 0.4991 - accuracy: 0.8244

Epoch 2/10

60000/60000 [=====] - 3s 53us/sample - loss: 0.3771 - accuracy: 0.8643

Epoch 3/10

60000/60000 [=====] - 3s 53us/sample - loss: 0.3381 - accuracy: 0.8775

Epoch 4/10

60000/60000 [=====] - 3s 53us/sample - loss: 0.3145 - accuracy: 0.8840

Epoch 5/10

60000/60000 [=====] - 3s 53us/sample - loss: 0.2950 - accuracy: 0.8908

Epoch 10/10

60000/60000 [=====] - 3s 53us/sample - loss: 0.2381 - accuracy: 0.9096



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Đánh giá mô hình
  - So sánh kết quả dự báo của mô hình với nhãn thật.

```
test_loss, test_acc =  
model.evaluate(test_images, test_labels, verbose=2)
```

```
print('\nTest accuracy:', test_acc)
```

10000/10000 - 1s - loss: 0.3548 - accuracy: 0.8802

Test accuracy: 0.8802



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- Tạo mô hình dự báo
  - Kết quả của tầng đầu ra là dữ liệu thô, không có hàm kích hoạt.
  - Để có thể sử dụng mô hình này để dự báo ảnh mới, ta cần thêm 1 tầng softmax nối vào cuối tầng đầu ra.

```
probability_model = tf.keras.Sequential([model,  
                                         tf.keras.layers.Softmax()])
```

```
predictions = probability_model.predict(test_images)
```



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

- predictions[0]

```
array([9.6798196e-09, 3.9551045e-12, 1.1966002e-09, 6.4597640e-11,  
       6.9501125e-09, 1.3103469e-04, 1.2785731e-07, 2.0190427e-01,  
       1.3739854e-09, 7.9796457e-01], dtype=float32)
```

- np.argmax(predictions[0])

9

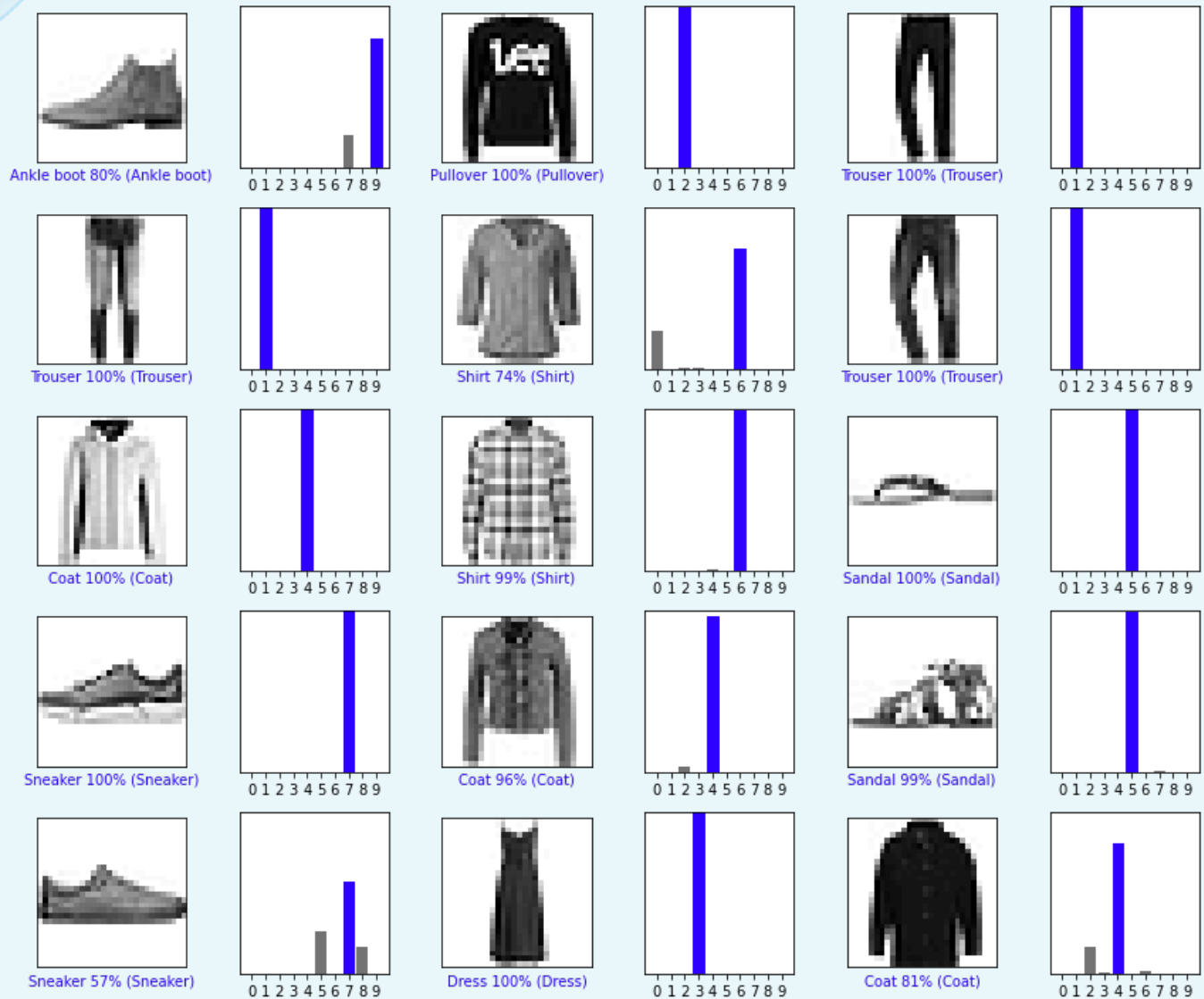
- test\_labels[0]

9



CANTHO UNIVERSITY

# Phân loại ảnh quần áo





# Phân loại ảnh quần áo

- Sử dụng mô hình đã huấn luyện
  - Lấy 1 ảnh từ tập kiểm tra.  
`img = test_images[1]`  
`print(img.shape)` (28, 28)
  - Thêm 1 chiều nữa cho giống với dữ liệu đầu vào  
`img = (np.expand_dims(img, 0))`  
`print(img.shape)` (1, 28, 28)
  - Dự báo nhãn  
`predictions_single = probability_model.predict(img)`  
`print(predictions_single)`



CANTHO UNIVERSITY

# Phân loại ảnh quần áo

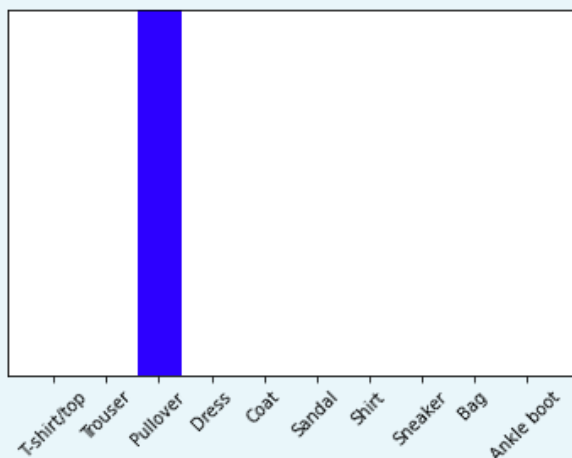
- Kết quả

```
[[1.1843847e-05 2.8502357e-11 9.9778062e-01 3.2734149e-10 2.0844834e-03  
3.5600198e-15 1.2303848e-04 1.4568713e-08 3.6617865e-11 5.2883337e-14]]
```

- Vẽ hình minh họa

```
plot_value_array(1, predictions_single[0],  
test_labels)
```

```
_ = plt.xticks(range(10), class_names, rotation=45)
```











CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Tập dữ liệu: IMDB
  - 50,000 bình luận về CSDL phim trên internet
  - Train - Test: 25,000 – 25,000
  - Lớp: positive, negative



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Cài đặt TF hub

```
pip install -q tensorflow-hub
```

```
pip install -q tfds-nightly
```

- Import

```
import numpy as np
```

```
import tensorflow as tf
```

```
import tensorflow_hub as hub
```

```
import tensorflow_datasets as tfds
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Download CSDL

```
train_data, validation_data, test_data = tfds.load(  
    name="imdb_reviews",  
    split=('train[:60%]', 'train[60%:]', 'test'),  
    as_supervised=True)
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Khám phá CSDL

```
train_examples_batch, train_labels_batch =  
next(iter(train_data.batch(10)))  
train_examples_batch
```

```
<tf.Tensor: shape=(10,), dtype=string, numpy=  
array([b"This was an absolutely terrible movie. Don't be lu  
      b'I have been known to fall asleep during films, but  
      b'Mann photographs the Alberta Rocky Mountains in a  
      b'This is the kind of film for a snowy Sunday aftern  
      b'As others have mentioned, all the women that go nu  
      b"This is a film which should be seen by anybody int  
      b'Okay, you have:<br /><br />Penelope Keith as Miss  
      b'The film is based on a genuine 1950s novel.<br /><  
      b'I really love the sexy action and sci-fi films of  
      b'Sure, this one isn\'t really a blockbuster, nor do  
dtype=object])>
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Khám phá CSDL

```
train_examples_batch, train_labels_batch =  
next(iter(train_data.batch(10)))  
train_examples_batch
```

```
<tf.Tensor: shape=(10,), dtype=string, numpy=  
array([b"This was an absolutely terrible movie. Don't be lu  
      b'I have been known to fall asleep during films, but  
      b'Mann photographs the Alberta Rocky Mountains in a  
      b'This is the kind of film for a snowy Sunday aftern  
      b'As others have mentioned, all the women that go nu  
      b"This is a film which should be seen by anybody int  
      b'Okay, you have:<br /><br />Penelope Keith as Miss  
      b'The film is based on a genuine 1950s novel.<br /><  
      b'I really love the sexy action and sci-fi films of  
      b'Sure, this one isn\'t really a blockbuster, nor do  
dtype=object])>
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Khám phá CSDL  
Nhãn của dữ liệu

```
train_labels_batch
```

```
<tf.Tensor: shape=(10,), dtype=int64, numpy=array([0, 0, 0, 1, 1, 1, 0, 0, 0, 0])>
```





CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Xây dựng mô hình
  - Tầng biểu diễn dữ liệu (TF Hub: [google/tf2-preview/gnews-swivel-20dim/1](https://tfhub.dev/google/tf2-preview/gnews-swivel-20dim/1))
  - Tầng ẩn (Dense), kích hoạt: RELU
  - Tầng đầu ra (Dense), kích hoạt: không



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Xây dựng mô hình

```
embedding = "https://tfhub.dev/google/tf2-  
preview/gnews-swivel-20dim/1"
```

```
hub_layer = hub.KerasLayer(embedding,  
input_shape=[],  
dtype=tf.string, trainable=True)  
hub_layer(train_examples_batch[:3])
```

```
<tf.Tensor: shape=(3, 20), dtype=float32, numpy=  
array([[ 1.765786 , -3.882232 ,  3.9134233 , -1.5557289 , -3.3362343 ,  
        -1.7357955 , -1.9954445 ,  1.2989551 ,  5.081598 , -1.1041286 ,  
        -2.0503852 , -0.72675157, -0.65675956,  0.24436149, -3.7208383 ,  
         2.0954835 ,  2.2969332 , -2.0689783 , -2.9489717 , -1.1315987 ],  
       [ 1.8804485 , -2.5852382 ,  3.4066997 ,  1.0982676 , -4.056685 ,
```



# Phân loại văn bản với TF Hub

- Xây dựng mô hình

```
model = tf.keras.Sequential()  
model.add(hub_layer)  
model.add(tf.keras.layers.Dense(16,  
activation='relu'))  
model.add(tf.keras.layers.Dense(1))
```

```
model.summary()
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Xây dựng mô hình

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
keras_layer (KerasLayer)	(None, 20)	400020
dense (Dense)	(None, 16)	336
dense_1 (Dense)	(None, 1)	17

```
Total params: 400,373  
Trainable params: 400,373  
Non-trainable params: 0
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Hàm loss

```
model.compile(optimizer='adam',  
loss=tf.keras.losses.BinaryCrossentropy(from_logits=True), metrics=['accuracy'])
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Huấn luyện

```
history =  
model.fit(train_data.shuffle(10000).batch(512),  
          epochs=20,  
          validation_data=validation_data.batch(512),  
          verbose=1)
```

```
Epoch 16/20  
30/30 [=====] - 2s 63ms/step - loss: 0.2294 - accuracy: 0.9099  
Epoch 17/20  
30/30 [=====] - 2s 62ms/step - loss: 0.2143 - accuracy: 0.9163  
Epoch 18/20  
30/30 [=====] - 2s 63ms/step - loss: 0.2008 - accuracy: 0.9209  
Epoch 19/20  
30/30 [=====] - 2s 62ms/step - loss: 0.1871 - accuracy: 0.9281  
Epoch 20/20  
30/30 [=====] - 2s 63ms/step - loss: 0.1749 - accuracy: 0.9337
```



CANTHO UNIVERSITY

# Phân loại văn bản với TF Hub

- Đánh giá mô hình

```
results = model.evaluate(test_data.batch(512),  
verbose=2)
```

```
for name, value in zip(model.metrics_names,  
results):
```

```
    print("%s: %.3f" % (name, value))
```

```
49/49 - 2s - loss: 0.3269 - accuracy: 0.8505  
loss: 0.327  
accuracy: 0.850
```



CẢM ƠN