Chương 7
Multi-layer Perceptron (2)

TS. Phạm Tuấn

ptuan@ute.udn.vn

Tensorflow

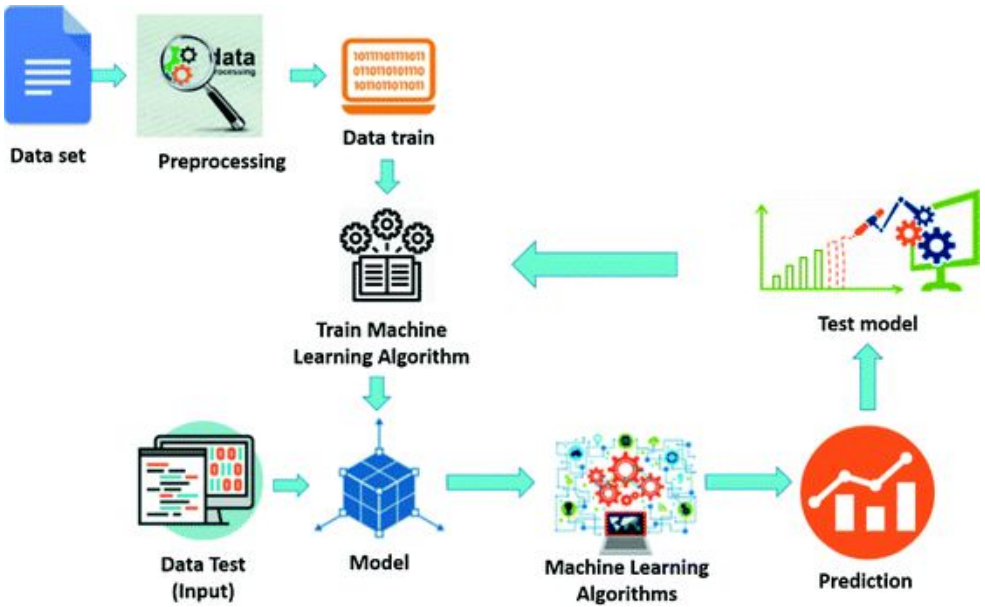
Tensorflow chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google.

1. **TensorBoard**: công cụ giúp minh họa các đồ thị tính toán (computational graph), sự thay đổi giá trị của các hàm tính toán (loss, accuracy,...) dưới dạng biểu đồ.
2. **TensorFlow Serving**: công cụ giúp triển khai các mô hình Machine Learning viết bằng TensorFlow thành một sản phẩm thực sự.
3. **Các API** giúp cho việc sử dụng TensorFlow dễ dàng hơn được phát triển bởi những nhà nghiên cứu về Machine Learning trên toàn thế giới (TensorFlow High Level API, TF-Slim, TensorFlow Object Detection API).

Xây dựng AI model

Cấu trúc của một chương trình AI sẽ gồm các bước sau:

1. Tiền xử lý dữ liệu.
2. Phác thảo AI model.
3. Train.
4. Theo dõi tiến trình train trên tập validation.
5. Lưu model tốt nhất trên tập validation.

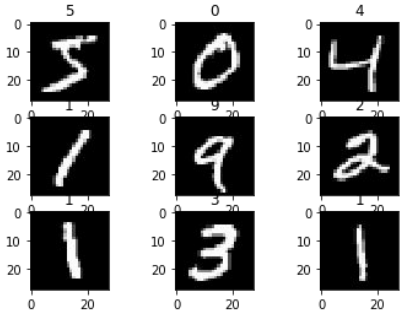


Thực hành với Tensorflow - dự đoán chữ số viết tay

```
# load thư viện và dữ liệu
import tensorflow as tf
(x_train, y_train), (x_test, y_test) =
tf.keras.datasets.mnist.load_data()

# kiểm tra số mẫu dữ liệu
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
```

```
(60000, 28, 28)
(60000,)
(10000, 28, 28)
(10000,)
```

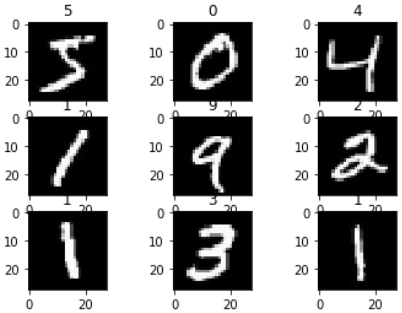
```
# vẽ một số dữ liệu ở tập train
import matplotlib.pyplot as plt
for i in range(9):
    plt.subplot(330+ 1 + i)
    plt.title(str(y_train[i]))
    plt.imshow(x_train[i], cmap=plt.get_cmap('gray'))
plt.show()
```



Thực hành với Tensorflow

```
# chuyển dữ liệu y_train từ label sang encode
# chuyển dữ liệu x_train về khoảng 0 và 1
from tensorflow.keras.utils import to_categorical
x_train, x_test = x_train / 255.0, x_test / 255.0
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
print(y_train[:9])
```

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



Thực hành với Tensorflow

```
#định nghĩa hàm loss
loss_fn = tf.keras.losses.CategoricalCrossentropy()

# định nghĩa thuật toán tối ưu
optimizer = tf.keras.optimizers.SGD(learning_rate=0.01)

# xây dựng mô hình
def create_model():
    model = tf.keras.models.Sequential([
        tf.keras.layers.Flatten(input_shape=(28, 28)),
        tf.keras.layers.Dense(128, activation='relu'),
    #   tf.keras.layers.Dropout(0.2),
        tf.keras.layers.Dense(256, activation='relu'),
        tf.keras.layers.Dense(256, activation='relu'),
        tf.keras.layers.Dense(10, activation = 'softmax')
    ])
    model.compile(optimizer=optimizer,
                  loss=loss_fn,
                  metrics=['accuracy'])

    return model
```

Thực hành với Tensorflow

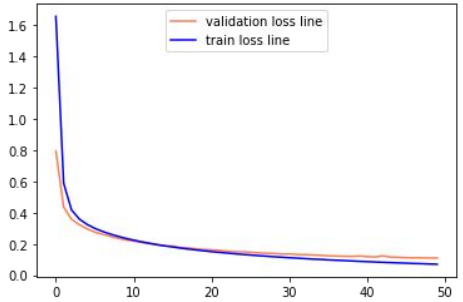
```
# tạo mô hình và in ra bảng tổng kết
model = create_model()
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 128)               100480
_________________________________________________________________
dense_1 (Dense)              (None, 256)               33024
_________________________________________________________________
dense_2 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_3 (Dense)              (None, 10)                2570
=================================================================
Total params: 201,866
Trainable params: 201,866
Non-trainable params: 0
_________________________________________________________________
```

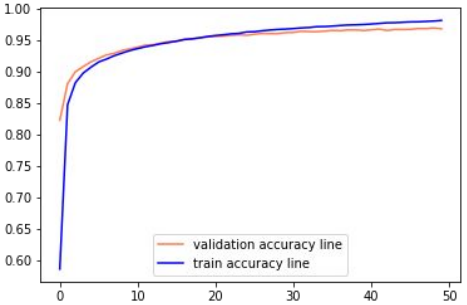
```
# xây dựng hàm lưu lại mô hình dựa theo loss của tập validation
weights_filepath = './weights/'
callback = tf.keras.callbacks.ModelCheckpoint( filepath =
weights_filepath, monitor='val_loss', verbose=1,
save_best_only=False, save_weights_only=False)

# bắt đầu training
his = model.fit(x_train, y_train, epochs = 50, batch_size = 128,
validation_split=0.2, callbacks=callback)
```

Thực hành với Tensorflow

```
# vẽ đường loss trên tập train và tập validation
plt.plot(his.history['val_loss'], c = 'coral', label = 'validation loss line')
plt.plot(his.history['loss'], c = 'blue', label = 'train loss line')
legend = plt.legend(loc='upper center')
plt.show()
```




```
# vẽ đường accuracy trên tập train và tập validation
plt.plot(his.history['val_accuracy'], c = 'coral', label = 'validation
accuracy line')
plt.plot(his.history['accuracy'], c = 'blue', label = 'train accuracy line')
legend = plt.legend(loc='lower center')
plt.show()
```



Thực hành với Tensorflow

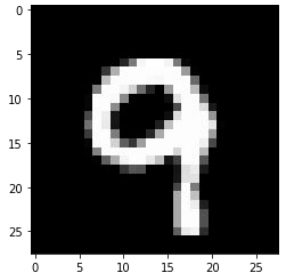
```
# Load file mô hình đã huấn luyện
model = create_model()
model = tf.keras.models.load_model('./weights/')

# Đánh giá mô hình trên tập test
loss, acc = model.evaluate(x_test, y_test, verbose=0)
print('loss tập test = ', loss,'| accuracy tập test = ', acc)
```

loss tập test = 0.09902060031890869
accuracy tập test = 0.9690999984741211

Thực hành với Tensorflow

```
# lấy 1 hình ảnh bất kỳ ở tập test và dự đoán
import numpy as np
input_image = x_test[99]
plt.imshow(input_image, cmap=plt.get_cmap('gray'))
print('shape của 1 bức ảnh', input_image.shape)
input_image = np.expand_dims(input_image, axis = 0)
print('shape phù hợp với mô hình là 3 chiều', input_image.shape)

output = model.predict(input_image)
print('số dự đoán là :', output.argmax())
```

shape của 1 bức ảnh (28, 28)
shape phù hợp với mô hình là 3 chiều (1, 28, 28)
số dự đoán là : 9



Bài tập nhóm số 7

Cho các bài toán sau:

1. Xây dựng mô hình MLP làm phép nhân với 2 số, mỗi số nằm trong dãy từ 1 đến 100.
2. Xây dựng mô hình MLP làm phép cộng với 2 số, mỗi số nằm trong dãy từ 1 đến 100.
3. Xây dựng mô hình MLP dự đoán nam nữ dựa trên tập data sau đây.
4. Xây dựng mô hình MLP dự đoán cân nặng dựa trên chiều cao ở tập data sau đây.
5. Xây dựng mô hình MLP dự đoán 3 loại hoa dựa trên tập data ở đây.

Trình bày các bước làm theo như bài học.

Any questions ?