C754

Introduction to TensorFlow 2.0

Agenda

- Keras API
  - Sequential model
  - Functional API
- Dataset API
- Low-level API

Functional API

- More flexible than tf.sequential
- Can handle models with multiple inputs or outputs that share some layers
- Create a model by **building graphs of layers**

Example: multiple inputs

Assume $y = x_1^2 + 2x_2^2 + \epsilon$, where $\epsilon \overset{iid}{\sim} N(0,1)$.

```
from tensorflow import keras
from tensorflow.keras import layers
x_1 = keras.Input(shape=(1,), name='x_1')
h1_1 = layers.Dense(100, activation='relu')(x_1)
h2_1 = layers.Dense(10, activation='relu')(h1_1)
o_1 = layers.Dense(1)(h2_1)
x_2 = keras.Input(shape=(1,), name='x_2')
h1_2 = layers.Dense(100, activation='relu')(x_2)
h2_2 = layers.Dense(10, activation='relu')(h1_2)
o_2 = layers.Dense(1)(h2_2)
outputs = layers.Add(name='y')([o_1, o_2])
model = keras.Model(inputs=[x_1, x_2], outputs=outputs)
```

Compile and train the model

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.MeanSquaredError())
```

```
history = model.fit(
    {'x_1': X[:,0], 'x_2': X[:,1]},
    {'y': y},
    batch_size=n, epochs=1000, verbose=0)
```

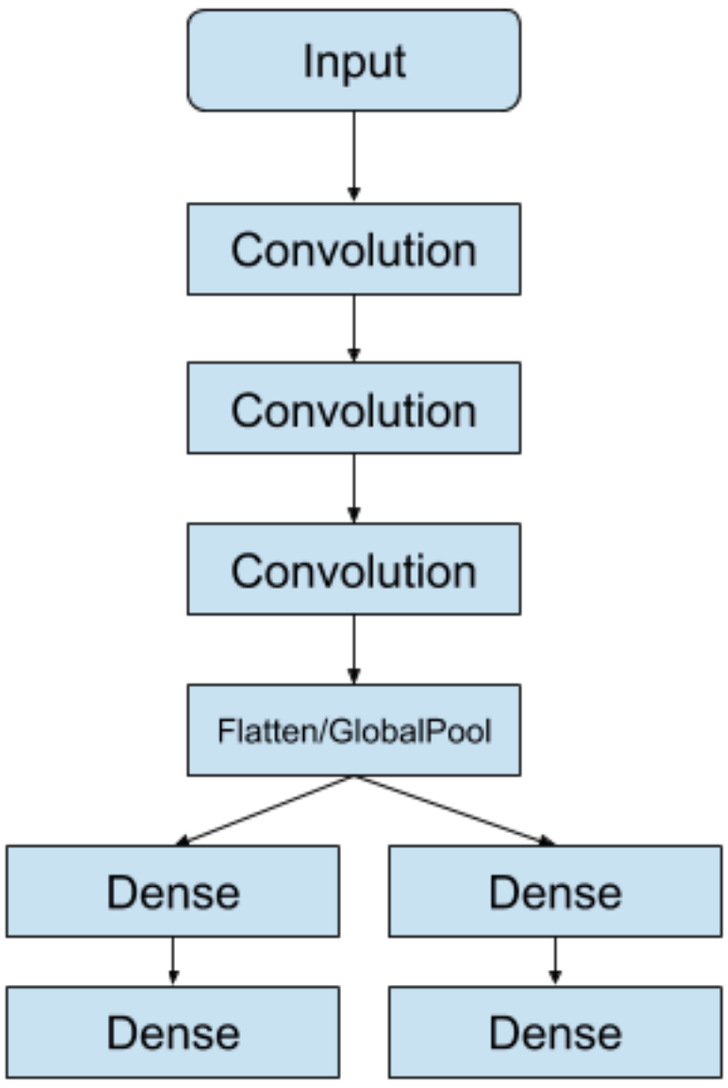
inputs與outputs需各自用dict打包給model.fit

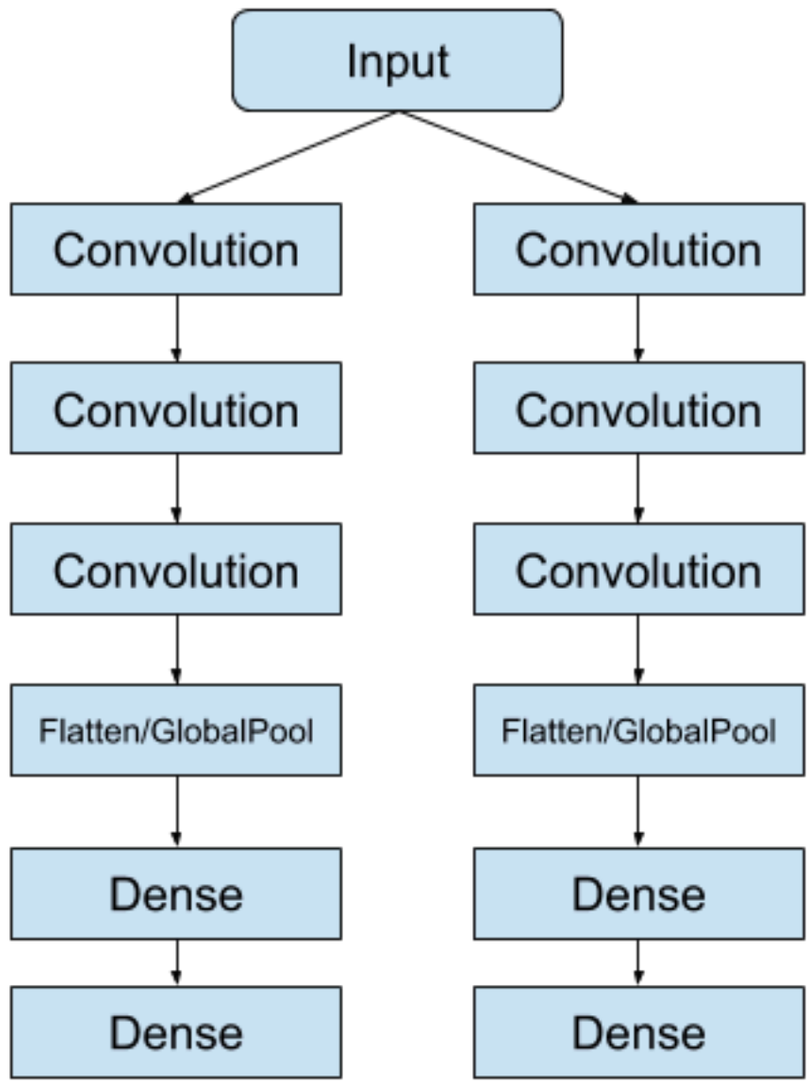
Predict by a functional API model

```
y_pred = model.predict({'x_1': X_test[:,0], 'x_2': X_test[:,1]})
mean_squared_error(y_test, y_pred)
```

inputs需用dict打包餵進model.predict

Example: multiple outputs





References

- 輕鬆學會 Google TensorFlow 2.0 人工智慧深度學習實作開發 (GitHub)

- Official documentation for tf.keras

- Guide to the Keras functional API in TensorFlow

Homework: multiclass logistic regression

1. Implement a multiclass logistic regression model by utilizing Keras functional API and apply it to the Iris dataset.

2. Evaluate the classification performances by cross-validation with some appropriate metrics.

3. Compare your results with those obtained by sklearn.linear_model.LogisticRegression with multi_class='multinomial'.