

Keras 簡介



2018/03/29 朱政安



簡介



- Keras是一個高層神經網路API，Keras由純Python編寫而成，並以Tensorflow、Theano以及CNTK為後端
 - 簡易和快速的原型設計（keras具有高度模塊化，極簡，和可擴充特性）
 - 支持CNN和RNN，或二者的結合
 - 無縫CPU和GPU切換



環境需求

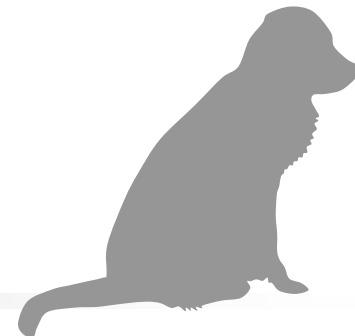


- Python (建議Anaconda)
- CUDA (GPU only)
- CuDNN (GPU only)

- Backend: Tensorflow



安裝



- 在CMD命令行或者Powershell中輸入：
 - pip install keras

- 驗證：

```
In[4]: import keras
Using TensorFlow backend.
```

- Keras 官網
- Keras 中文文檔



Keras Example



- VGG-16 model [1]:
- 5層 conv. layers
- 3層 fully connected layers



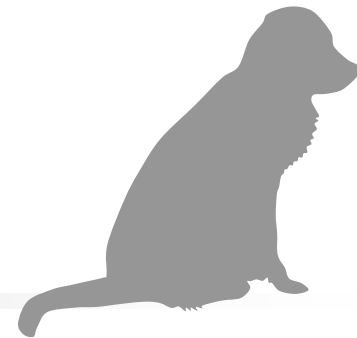
Keras Example



- Data: MNIST dataset
- 28*28的灰階手寫數字圖片
- 60000訓練資料
- 10000測試資料



Keras Example



- Data process:

導入函式庫

```
import numpy as np
from keras.datasets import mnist
from keras.utils import np_utils
```

--Keras內建MNIST資料庫

資料維度為(batch, channel, rows, columns)

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
# data pre-processing
X_train = X_train.reshape(-1, 1, 28, 28)/255.
X_test = X_test.reshape(-1, 1, 28, 28)/255.
y_train = np_utils.to_categorical(y_train, num_classes=10)
y_test = np_utils.to_categorical(y_test, num_classes=10)
```

--導入訓練測試資料

--訓練資料normalize

--測試資料normalize

--訓練資料標籤轉換

--測試資料標籤轉換

1 – [1 0 0 0 0 0 0 0 0 0]

5 – [0 0 0 0 1 0 0 0 0 0]

One-hot encoding



Keras Example



- Model: 導入函式庫

```
from keras.models import Sequential
from keras.layers import Dense, Activation, Convolution2D, MaxPooling2D, Flatten, Dropout
from keras.optimizers import Adam, sgd
```

- 一層一層建立模型

```
model = Sequential()
```



Keras Example



- 建立 VGG16: conv1
 - Conv1_1
 - Relu
 - Conv1_2
 - Relu
 - MaxPooling



Keras Example



- Conv1_1 (Convolution2D):

```
model.add(Convolution2D(  
    batch_input_shape=(None, 1, 28, 28),  
    filters=64,  
    kernel_size=3,  
    strides=1,  
    padding='same',  
    data_format='channels_first',  
))  
model.add(Activation('relu'))
```

- 第一層要定義輸入資料
- 本層有64個filters，也就是有64個output
- 本層filters大小為3*3
- 本層filters每次位移為1
- 有zero padding
- 資料維度是[channel, rows, columns]
就要加這句
- 本層輸出結果套用relu激勵函數
由於是使用Sequential來建立模型
所以會默認每一層的輸出
為下一層的輸入
所以不用在特別定義輸入資料



Keras Example



- Conv1_2:

```
model.add(Convolution2D(  
    batch_input_shape=(None, 1, 28, 28),  
    filters=64,  
    kernel_size=3,  
    strides=1,  
    padding='same',  
    data_format='channels_first',  
))  
model.add(Activation('relu'))
```

Keras Example



- Pooling1 (MaxPooling2D):

```
model.add(MaxPooling2D(  
    pool_size=2,  
    strides=2,  
    padding='same',  
    data_format='channels_first',  
))
```

--本層pooling filters大小為2*2

--本層pooling filters位移為2

Keras Example



- Conv5_1:

```
model.add(Convolution2D(  
    filters=512,  
    kernel_size=3,  
    strides=1,  
    padding='same',  
    data_format='channels_first',  
))  
model.add(Activation('relu'))
```

Keras Example



- Conv5_2:

```
model.add(Convolution2D(  
    filters=512,  
    kernel_size=3,  
    strides=1,  
    padding='same',  
    data_format='channels_first',  
))  
model.add(Activation('relu'))
```

Keras Example



- Conv5_3:

```
model.add(Convolution2D(  
    filters=512,  
    kernel_size=3,  
    strides=1,  
    padding='same',  
    data_format='channels_first',  
))  
model.add(Activation('relu'))
```

Keras Example



- Pooling5 :

```
model.add(MaxPooling2D(  
    pool_size=2,  
    strides=2,  
    padding='same',  
    data_format='channels_first',  
))
```


Keras Example



- Fc6 (Dense) :

```
model.add(Flatten())  
model.add(Dense(4096))  
model.add(Activation('relu'))  
model.add(Dropout(0.5))
```

--將conv層的結果壓成一維

--本層有4096個神經元

--每個神經元有50%機率丟棄資訊

Keras Example



- Fc7 : 不用 Flatten

```
model.add(Dense(4096))  
model.add(Activation('relu'))  
model.add(Dropout(0.5))
```

Keras Example



- fc8 :

```
model.add(Dense(10))  
model.add(Activation('softmax'))
```

--資料類別數量

- 本層和分類有關

Keras Example



- 激活模型：

```
sgd = sgd(lr=0.1, decay=1e-4, momentum=0.9)
model.compile(optimizer=sgd,
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

--設置優化器

--選擇loss function

--放入需要計算的



Keras Example



- 開始訓練模型：

```
model.fit(X_train, y_train, epochs=1, batch_size=64,)
```

- 輸入訓練資料和標籤
- 訓練1個epoch (全部訓練資料訓練一次)
- 64筆資料為1個batch



Keras Example



- 測試模型：

```
score = model.evaluate(X_test, y_test)
print('\ntest loss: ', score[0])
print('\ntest accuracy: ', score[1])
```

--將測試資料進行測試

--印出loss值

--印出accuracy