데이터분석도 자동화가 될 수 있을까?

Auto-ML

많은 사람들이 AI 시대에서 가장 두려운 것 중 하나는 일자리를 잃는 것이라고 말한다. (나는 인간의 영역이 반복 업무에서 보다 창조적인 영역으로 이동하는 것이라고 믿고 싶다.) 현재 AI는 단순 반복 작업 뿐만 아니라 그림 그리기, 글 쓰기 등 인간의 창조적인 영역이라고 일컬어지는 분야까지 넓게 발전되고 있지만, 자동화의 첫 번째 단계는 단순 반복 작업부터 시작될 것이다.

아이러니하게도? 이러한 자동화를 가능하게 하는 머신러닝 조차도 자동화 할 수 있는 여러 방법들이 연구되고 있다. 머신러닝 자동화를 살펴보기에 앞서, 머신러닝 파이프라인을 살펴보자.

데이터분석을 이용하여 문제를 해결할 때 다음과 같은 프로세스를 반복적으로 거친다.

- 데이터 수집
- 데이터 전처리
- 데이터 탐색
- 모델 학습
- 모델 평가
- 결과 분석
- 파라미터 튜닝
- 모델 평가 및 분석

데이터 전처리 및 탐색부터 파라미터 튜닝까지 이 작업들은 굉장히 광범위하고 반복적으로 일어난다. 한 스텝 단계에서의 변화가 이후 파이프라인 프로세스에 영향을 줄 수 있기 때문에 경우의 수도 많아지고, 많은 시간이 소요될 수 밖에 없다.

더욱 어려운 점은, 이것이 최선의 결과인지, 더 좋은 모델은 없는지, 어느 시점에서 분석을 멈춰야 하는지 알기 어렵다는 것이다. 이 과정에서 때로 데이터분석가들은 돌고 도는 자신만의 챗바퀴 함정에 빠지게 될수도 있다.

데이터분석가들은 문제를 파악하고, 데이터를 탐색하며 정제하는데 많은 시간을 쏟는다. 마감시간은 늘 빠듯하기 때문에, 모델 튜닝이나 정확도 개선을 위한 작업을 할 시간이 부족할 수 밖에 없다. 여기서 **AutoML의 아이디어**가 나왔다. **다양한 알고리즘 및 다양한 하이퍼 파라미터를 구성하여 수많은 모델을 생성하고 성능과 정확도를 비교하는 것을 자동화 하는 것이다.**

이러한 작업을 위해 개발된 파이썬 라이브러리는 다음과 같다.

- Auto-Sklearn
- TPOT
- Auto-Keras
- H2O.ai
- Google’s AutoML

MNIST 데이터 셋을 이용하여 이 중 AutoKeras와 Keras를 비교해보고자 한다.

MNIST Classification with Keras

이 중 AutoKeras 사용법을 알아보기 위해 Keras 기반 MNIST 분류를 아래와 같이 수행해보았다.

```
'''Trains a simple convnet on the MNIST dataset.
Gets to 99.25% test accuracy after 12 epochs
(there is still a lot of margin for parameter tuning).
16 seconds per epoch on a GRID K520 GPU.
'''

from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

batch_size = 128
num_classes = 10
epochs = 12

# input image dimensions
img_rows, img_cols = 28, 28

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = mnist.load_data()

if K.image_data_format() == 'channels_first':
    x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
    x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
    input_shape = (1, img_rows, img_cols)
else:
    x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
    x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
    input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print('x_train shape:', x_train.shape)
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')
```

```
# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
Using TensorFlow backend.


Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
(https://s3.amazonaws.com/img-datasets/mnist.npz)
11493376/11490434 [==============================] - 6s 0us/step
x_train shape: (60000, 28, 28, 1)
60000 train samples
10000 test samples
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
60000/60000 [==============================] - 138s 2ms/step - loss: 0.2622 - acc:
0.9205 - val_loss: 0.0573 - val_acc: 0.9822
Epoch 2/12
60000/60000 [==============================] - 138s 2ms/step - loss: 0.0875 - acc:
0.9744 - val_loss: 0.0393 - val_acc: 0.9866
Epoch 3/12
60000/60000 [==============================] - 138s 2ms/step - loss: 0.0622 - acc:
0.9815 - val_loss: 0.0337 - val_acc: 0.9874
Epoch 4/12
60000/60000 [==============================] - 144s 2ms/step - loss: 0.0553 - acc:
0.9835 - val_loss: 0.0301 - val_acc: 0.9894
Epoch 5/12
60000/60000 [==============================] - 145s 2ms/step - loss: 0.0457 - acc:
0.9866 - val_loss: 0.0336 - val_acc: 0.9887
Epoch 6/12
60000/60000 [==============================] - 135s 2ms/step - loss: 0.0403 - acc:
0.9879 - val_loss: 0.0293 - val_acc: 0.9912
Epoch 7/12
60000/60000 [==============================] - 140s 2ms/step - loss: 0.0355 - acc:
0.9892 - val_loss: 0.0269 - val_acc: 0.9911
Epoch 8/12
60000/60000 [==============================] - 138s 2ms/step - loss: 0.0337 - acc:
0.9894 - val_loss: 0.0268 - val_acc: 0.9914
Epoch 9/12
60000/60000 [==============================] - 142s 2ms/step - loss: 0.0309 - acc:
0.9903 - val_loss: 0.0275 - val_acc: 0.9917
Epoch 10/12
60000/60000 [==============================] - 139s 2ms/step - loss: 0.0278 - acc:
0.9914 - val_loss: 0.0280 - val_acc: 0.9920
Epoch 11/12
```

```
60000/60000 [==============================] - 139s 2ms/step - loss: 0.0284 - acc:
0.9914 - val_loss: 0.0265 - val_acc: 0.9921
Epoch 12/12
60000/60000 [==============================] - 138s 2ms/step - loss: 0.0271 - acc:
0.9917 - val_loss: 0.0318 - val_acc: 0.9912
Test loss: 0.031818649467292154
Test accuracy: 0.9912
```

MNIST Classification with AutoKeras

위의 손글씨 분류를 위한 keras 예제는 tensorflow, pytorch 버전에 비해 굉장히 간단하게 작성된 것이다. 하지만 AutoKeras를 사용하면 실행코드가 단 4줄로 표현된다. 코드양은 획기적으로 줄었지만 내부적으로 여러 모델과 파라미터 등을 비교하기 때문에 소요시간은 길다.

```
from keras.datasets import mnist
from autokeras import ImageClassifier

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.reshape(x_train.shape + (1,)) # (1,) denotes the channles which
is 1 in this case
x_test = x_test.reshape(x_test.shape + (1,)) # (1,) denotes the channles which is
1 in this case
```

```
clf = ImageClassifier(verbose=True, augment=False)

clf.fit(x_train, y_train, time_limit=12 * 60 * 60)
clf.final_fit(x_train, y_train, x_test, y_test, retrain=True)
y = clf.evaluate(x_test, y_test)

print(y * 100)
print("end")
```

```
Preprocessing the images.
Preprocessing finished.

Initializing search.
Initialization finished.


+--------------------------------------------+
|               Training model 0             |
+--------------------------------------------+

No loss decrease after 5 epochs.


Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           0            |  0.19162312000989914   |         0.9868         |
+--------------------------------------------------------------------------+



+--------------------------------------------+
|               Training model 1             |
+--------------------------------------------+

No loss decrease after 5 epochs.


+--------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+--------------------------------------------------------------------------+
|           0            |              to_wider_model 2 64                |
+--------------------------------------------------------------------------+

Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           1            |   0.118541076220572    |         0.9916         |
+--------------------------------------------------------------------------+
```

```
+-----------------------------------------------+
|               Training model 2                |
+-----------------------------------------------+

No loss decrease after 5 epochs.


+--------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+--------------------------------------------------------------------------+
|           1            |    to_deeper_model 43 Conv2d(256, 256, 5, 1)    |
+--------------------------------------------------------------------------+

Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           2            |  0.11978653594851493   |         0.9904         |
+--------------------------------------------------------------------------+



+-----------------------------------------------+
|               Training model 3                |
+-----------------------------------------------+

No loss decrease after 5 epochs.


+--------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+--------------------------------------------------------------------------+
|           1            |     to_deeper_model 67 BatchNormalization2d     |
+--------------------------------------------------------------------------+

Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           3            |  0.15989562943577768   |   0.9912000000000001   |
```

```
+----------------------------------------------------------------------------+



+--------------------------------------------+
|               Training model 4             |
+--------------------------------------------+

No loss decrease after 5 epochs.



+----------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+----------------------------------------------------------------------------+
|           1            |             to_deeper_model 38 ReLU             |
+----------------------------------------------------------------------------+

Saving model.
+----------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+----------------------------------------------------------------------------+
|           4            |  0.15264498740434645   |   0.9907999999999999   |
+----------------------------------------------------------------------------+



+--------------------------------------------+
|               Training model 5             |
+--------------------------------------------+

No loss decrease after 5 epochs.



+----------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+----------------------------------------------------------------------------+
|           1            |      to_deeper_model 3 BatchNormalization2d     |
+----------------------------------------------------------------------------+

Saving model.
+----------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+----------------------------------------------------------------------------+
```

```
|           5            |  0.15766185969114305   |         0.9896         |
+------------------------------------------------------------------------+



+---------------------------------------------+
|               Training model 6              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+------------------------------------------------------------------------+
|    Father Model ID     |                 Added Operation               |
+------------------------------------------------------------------------+
|           1            |            to_concat_skip_model 1 59          |
+------------------------------------------------------------------------+

Saving model.
+------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value     |
+------------------------------------------------------------------------+
|           6            |   0.1667788177728653   |   0.9932000000000001  |
+------------------------------------------------------------------------+



+---------------------------------------------+
|               Training model 7              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+------------------------------------------------------------------------+
|    Father Model ID     |                 Added Operation               |
+------------------------------------------------------------------------+
|           6            |     to_deeper_model 7 Conv2d(64, 64, 3, 1)    |
+------------------------------------------------------------------------+

Saving model.
+------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value     |
```

```
+--------------------------------------------------------------------------+
|           7            |  0.11798861380666495   |   0.9932000000000001   |
+--------------------------------------------------------------------------+



+---------------------------------------------+
|               Training model 8              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+--------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+--------------------------------------------------------------------------+
|           6            |    to_deeper_model 42 Conv2d(256, 256, 5, 1)    |
+--------------------------------------------------------------------------+

Saving model.
+--------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+--------------------------------------------------------------------------+
|           8            |  0.11608820855617523   |         0.9916         |
+--------------------------------------------------------------------------+



+---------------------------------------------+
|               Training model 9              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+--------------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                  |
+--------------------------------------------------------------------------+
|                        |     to_deeper_model 21 BatchNormalization2d     |
|           6            |              to_wider_model 31 128              |
+--------------------------------------------------------------------------+

Saving model.
```

```
+-----------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------+
|           9            |  0.14258270002901555   |   0.9895999999999999   |
+-----------------------------------------------------------------------+



+-----------------------------------------------+
|              Training model 10                |
+-----------------------------------------------+

No loss decrease after 5 epochs.



+-----------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                |
+-----------------------------------------------------------------------+
|                        |            to_deeper_model 43 ReLU            |
|           6            |             to_wider_model 31 128             |
+-----------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------+
|           10           |  0.15620764940977097   |         0.9924         |
+-----------------------------------------------------------------------+



+-----------------------------------------------+
|              Training model 11                |
+-----------------------------------------------+

No loss decrease after 5 epochs.



+-----------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                |
+-----------------------------------------------------------------------+
|                        |            to_deeper_model 31 ReLU            |
|           6            |             to_wider_model 58 512             |
```

```
+-----------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------+
|           11           |  0.11643899343907833   |   0.9932000000000001   |
+-----------------------------------------------------------------------+



+---------------------------------------------+
|              Training model 12              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+-----------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                |
+-----------------------------------------------------------------------+
|                        |     to_deeper_model 3 BatchNormalization2d    |
|           6            |              to_wider_model 31 128            |
+-----------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------+
|           12           |  0.15061972439289092   |         0.9904         |
+-----------------------------------------------------------------------+



+---------------------------------------------+
|              Training model 13              |
+---------------------------------------------+



No loss decrease after 5 epochs.

+-----------------------------------------------------------------------+
|    Father Model ID     |                Added Operation                |
```

```
+-----------------------------------------------------------------------------+
|           6           |           to_concat_skip_model 39 68                |
+-----------------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------------+
|        Model ID       |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------------+
|           13          |   0.1234286069869995   |         0.9936         |
+-----------------------------------------------------------------------------+



+---------------------------------------------+
|              Training model 14              |
+---------------------------------------------+

No loss decrease after 5 epochs.



+-----------------------------------------------------------------------------+
|    Father Model ID    |                Added Operation                      |
+-----------------------------------------------------------------------------+
|                       |    to_deeper_model 33 Conv2d(256, 256, 5, 1)        |
|           13          |              to_wider_model 34 128                  |
+-----------------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------------+
|        Model ID       |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------------+
|           14          |  0.11035485975444317   |   0.9928000000000001   |
+-----------------------------------------------------------------------------+



+---------------------------------------------+
|              Training model 15              |
+---------------------------------------------+

No loss decrease after 5 epochs.
```

```
+-----------------------------------------------------------------------------+
|    Father Model ID     |                 Added Operation                    |
+-----------------------------------------------------------------------------+
|                        |    to_deeper_model 61 Conv2d(512, 512, 1, 1)     |
|           13           |               to_wider_model 13 64                 |
+-----------------------------------------------------------------------------+

Saving model.
+-----------------------------------------------------------------------------+
|        Model ID        |          Loss          |      Metric Value      |
+-----------------------------------------------------------------------------+
|           15           |   0.1063856065273285   |         0.994          |
+-----------------------------------------------------------------------------+
```

MNIST 분류 모델 정확도는 Keras가 0.9912, AutoKeras가 0.994로 AutoKeras 정확도가 좀 더 높다. AutoKeras 진행 과정을 보면 Father Model을 두고 거기에 added_operation을 적용해 모델 정확도를 높여가는 방식이다. 딥러닝 네트워크 구조를 바꾼다거나, deeper / wider / concat 등으로 모델에 변형을 준다. 위와 같이 model 15번에서 끝난 것이 아니라 계속 돌고 있는 상태인데 (15번까지 도는데 12시간 이상 소요) 너무 시간이 오래 걸려서 중간에 멈춘 결과이다. MNIST 데이터를 이용하여 끝까지 돌리면 0.998%의 높은 정확도가 나온다고 한다.

단상

사실 예전부터 데이터모델링에 대한 자동화 이야기는 줄곧 화자되어 왔다. 하지만 도메인을 이해해야하고, 전처리 및 feature engineering 등에 대한 기준을 만들고, 어떤 아이디어를 적용할지 의사결정을 하는 것은 자동화 할 수 없는 부분이기 때문에 회의적이었다.

위의 MNIST 예제를 보면 코드양이 단 4줄로 바뀌기도 하고 정확도가 0.9912 -> 0.994(0.998)로 높아지긴 했지만, 과연 실무에서 사용되는 dirty-data에서도 AutoKeras가 제대로 동작될지는 아직 의심스럽다.

하지만 모델링 방법 선택 (어떤 분류기를 사용할 것인지 등), 하이퍼파라미터 튜닝, 여러가지 딥러닝 네트워크 시도 등은 경우의 수가 너무 많아 사람이 직접 적용하고 결과 비교하기에는 너무 많은 시간이 소요되기 때문에, 이 time-consuming 되는 프로세스 부분에 Auto-ML 방식을 사용하면 지쳐가는 데이터 사이언티스트를 도울 수 있지 않을까.

하지만 Auto-ML 방식만으로 결과가 좋아지리라 기대하긴 어렵다. 전제조건으로 좋은 컴퓨팅 환경이 뒷받침 되야 할 것이고 (하이퍼파라미터 튜닝, added operation으로 시간이 정말 많이 소요), 특정 파이프라인만 Auto-ML 방식을 쓰는 hybrid한 형태가 적합할 것으로 보인다. 추후에 정제되지 않은 데이터에도 적용해보고 사람이 손수 돌린 결과와 정확도 비교를 해보고 싶다.

Reference

- https://www.datacamp.com/community/tutorials/automated-machine-learning-auto-keras (https://www.datacamp.com/community/tutorials/automated-machine-learning-auto-keras)

태그: AutoKeras Data Analysis

카테고리: Data Analysis

업데이트: January 14, 2019