Технологии и машинное обучение

Штейнберг Роман Борисович
ИММиКН ЮФУ
CVisionLab

Это ваши технологии?

SciKit-Learn

http://scikit-learn.org/

- Содержит очень много алгоритмов (модели, предобработка ...)
- Представлены реализации из различных библиотек
- Есть возможность загрузки тренировочных датаестов
- Pipeline

SVM vs Forests

- SVM позволяет использовать ядра
- DF позволяет строить очень гибкую систему принятия решений

SVM и DF в SciKit-Learn

libLinear

libSVM

xgboost

Caffe

http://caffe.berkeleyvision.org/

- Сверточные нейронные сети
- C++ реализация
- Cuda backend
- Настройка через protobuf-файлы
- Высокие показатели: 72 миллиона изображений в сутки

Сравнение на бенчмарках: https://github.com/soumith/convnet-benchmarks

Как повысить эффективность своей работы?

- Предобработка с сохранением результатов
- Распараллеливание
- Использование графического ускорителя

Кроме того,

- Визуализация
- Тестирование

hdf5 и pickle

- pickle позволяет сохранить и восстановить данные в одну строчку
- hdf5 позволяет эффективно хранить данные

Hierarchical Data Format (hdf5)

https://www.hdfgroup.org/HDF5/

- API на C, C++, Python (h5py), Fortran90, Java …
- Поддержка параллельного ввода/вывода (parallel I/O)
- Поддержка поизвольного доступа (random access)
- Индексация с помощью B-деревьев, при этом данные храняться в последовательных ячейках памяти (straightforward arrays)
- Механизм хранения проще и быстрее, чем SQL (star schema)

Еще о hdf5

- Знакомая структура хранения, как у файловых систем
- Хранение частями (chunks)
- Сжатие (compress)

Недостатки (информация возможно устарела)

- Нет utf-8
- Проблемы с соблюдаение стандарта
- Удаление объектов не ведет к уменьшению размера датасета

Пример использования hdf5

```
import h5py, numpy, pylab

dataset = h5py.File('data.h5f', 'r')
images = dataset['images']  # almost numpy array
pylab.imshow(images[0])
dataset.close()
```

Torch 7, Theano, TensorFlow

http://torch.ch/

http://deeplearning.net/software/theano/

https://www.tensorflow.org/

- Большинство мнений найденных в Интернете, говорят что Theano быстрее TensorFlow, в том числе https://goo.gl/kP7ioP .
- У Torch 7 - Lua, а у TensorFlow - Jeff Dean.
- TensorFlow подружился с CodinGame

Ответ Microsoft - CNTK

https://www.cntk.ai/

https://github.com/Microsoft/CNTK - здесь есть сравнение технологий

CNTK - это Computational Network Toolkit

- Использует до 8 GPU, тогда как Theano только 1.
- Даже multi-machine-multi-GPU scenarios

Пример использования Theano

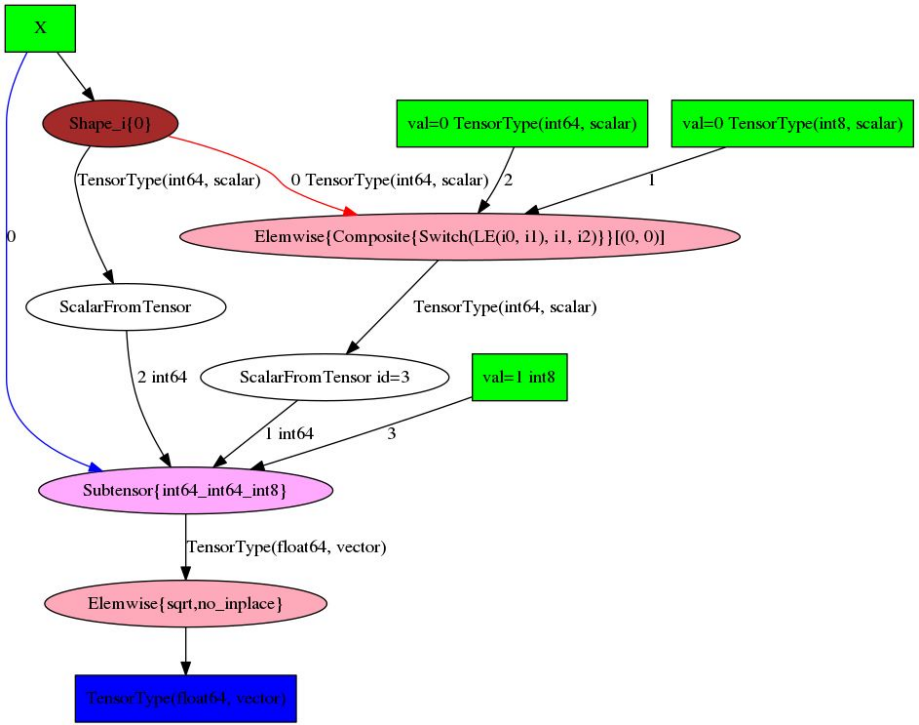
```
import theano, theano.tensor as T, numpy as np

# define tensor variable
X = T.vector("X")
results, updates = theano.scan(lambda x_i: T.sqrt(x_i), sequences=[X])
compute_norm_lines = theano.function(inputs=[X], outputs=results)

# test value
x = np.arange(1, 6, dtype=theano.config.floatX)
print(compute_norm_lines(x))
print(np.sqrt(x))

# visualize graph
theano.printing.pydotprint(compute_norm_lines, outfile='/home/rl/theano.png', scan_graphs=True,
var_with_name_simple=True)
```

Визуализация графа вычислений



Keras и Lasagne

http://keras.io/

http://lasagne.readthedocs.io/en/latest/

Keras ставит во главу угла простоту, а Lasagne - прозрачность.

Создание CNN с помощью Keras делается потрясающе просто!

Пример CNN на Keras

```
from keras.models import Sequential
from keras.layers.core import Dense, Activation
from keras.layers.convolutional import Convolution2D, MaxPooling2D

def create_my_model(data_shape):
        nb_classes = 10
        model = Sequential()
        model.add(Convolution2D(32, 3, 3, input_shape=data_shape))
        model.add(Activation('relu'))
        model.add(Convolution2D(32, 3, 3))
        model.add(Activation('relu'))
        model.add(MaxPooling2D(pool_size=(2, 2)))
        model.add(Dropout(0.25))
        model.add(Dense(nb_classes))
        model.add(Activation('softmax'))
        model.compile(loss='mse', optimizer='sgd', metrics=['accuracy'])
        return model
```