

Подготовка к спринту

- Если аккаунта на `github.com` нет, зарегистрироваться
- Форкнуть репозиторий <https://github.com/catboost/catboost>
- Склонировать форкнутую версию на разработческую машину
- Запустить сборку CLI версии (смотри инструкции для [linux/macos](#) и [windows](#))
- Запустить сборку Python package (смотри инструкции для [linux/macos](#) и [windows](#))
- Для Linux и MacOS можно сгенерировать `clion/qt` проект с помощью `./ya ide` На Windows можно пользоваться доступным в папке `msvs` решением

Яндекс

Открытый код в Яндексе.
Спринты по CatBoost и
ClickHouse



Введение в разработку CatBoost

Станислав Кириллов, ведущий разработчик CatBoost

Введение в разработку ClickHouse

Алексей Миловидов, разработчик ClickHouse

CatBoost

catboost / catboost

Unwatch

174

Unstar

3,632

Fork

518

<> Code

Issues 93

Pull requests 4

Insights

Settings

A fast, scalable, high performance Gradient Boosting on Decision Trees library, used for ranking, classification, regression and other machine learning tasks for Python, R, Java, C++. Supports computation on CPU and GPU. <https://catboost.ai>

Edit

machine-learning

decision-trees

gradient-boosting

gbm

gbdt

python

r

kaggle

gpu-computing

catboost

tutorial

categorical-features

gpu

coreml

data-science

big-data

cuda

data-mining

Manage topics

C++ 80.6%

Python 12.4%

Cuda 4.8%

R 0.7%

Makefile 0.6%

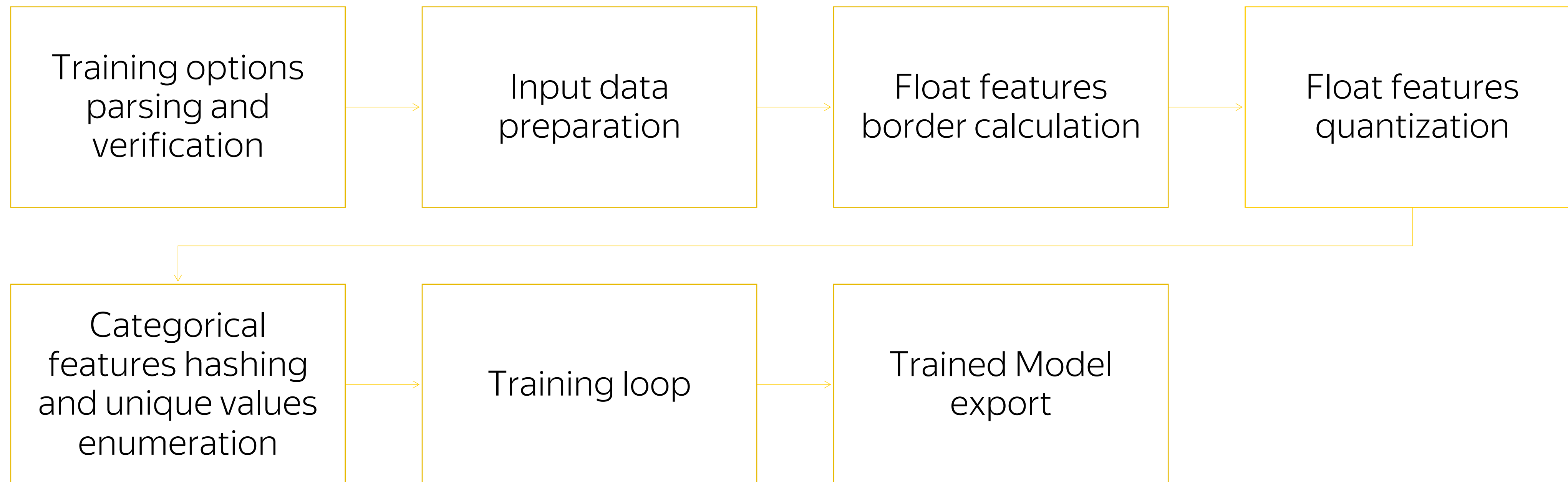
Java 0.4%

Other 0.5%

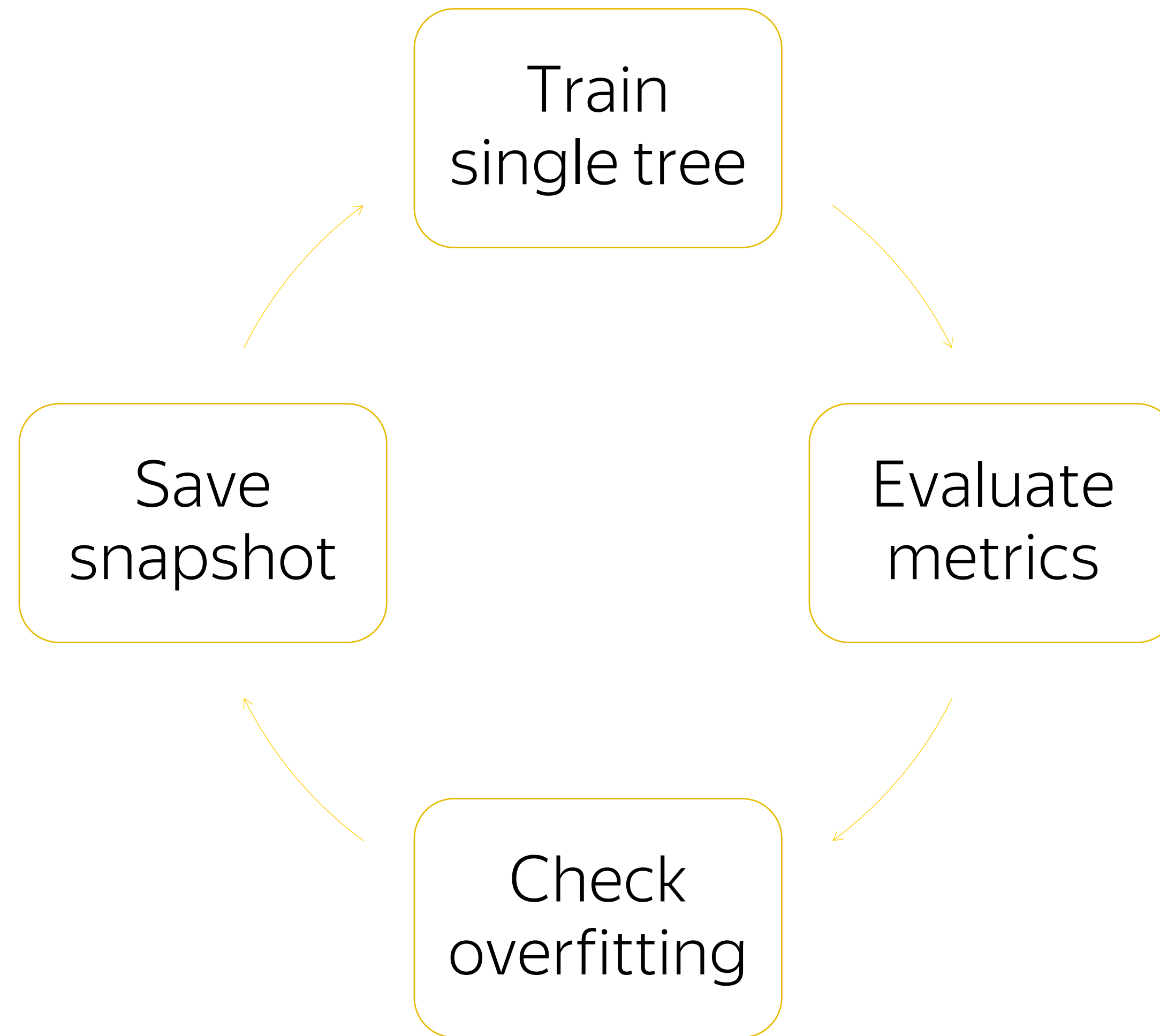
Как Catboost учится



Процесс обучения




Главный цикл обучения



Подбор одного дерева



Важные места в коде (обучение на CPU)



- catboost/libs/algo/greedy_tensor_search.cpp
- catboost/libs/algo/train.cpp
- catboost/libs/algo/approx_calcer.cpp
- catboost/libs/algo/score_calcer.cpp
- catboost/libs/train_lib/train_model.cpp
- catboost/libs/model/model.h
- catboost/python_package/catboost/_catboost.pyx
- catboost/libs/options

C++ в стиле Yandex



Структура репозитория

`util/` - библиотека системных примитивов: контейнеры, файловая система, строки, кодировки, потоки

`library/` - библиотеки общего пользования

`catboost/` - главная папка с кодом проекта CatBoost

Примитивы C++ (умные указатели)

util/generic/ptr.h

- › `THolder<T>` (аналог `std::unique_ptr<T>`)
- › `T(Atomic|Simple)SharedPtr<T>` (аналог `std::shared_ptr<T>`)
- › `TIntrusivePtr<T>` - хранит указатель на объекты-наследники класса **TRefCounted** экономия на аллокации control-block

Примитивы C++ (потоки ввода/вывода)

util/stream/*

- › `IInputStream` – базовый класс потоков ввода (operator >>)
- › `IOutputStream` – базовый класс потоков вывода (operator <<)
- › `Cin` – аналог `std::cin`
- › `Cout` – аналог `std::cout`
- › `Cerr` – аналог `std::cerr`
- › `Endl` – аналог `std::endl`

Примитивы C++ (работа с файлами)

util/stream/file.h

- › `TInputFile` – аналог `std::ifstream`
- › `TOutputFile` – аналог `std::ofstream`

util/system/fs.h

- › `NFs::Exists()` – проверка наличия файла/директории
- › `NFs::Copy()` – копировать файл

Примитивы C++ (контейнеры)

util/generic/vector.h

- › `TVector<T>` – наследник `std::vector<T>`

util/generic/hash.h

- › `THashMap<T>` – эквивалент `std::unordered_map<T>`
- › `THashSet<T>` – эквивалент `std::unordered_set<T>`

util/generic/set.h + util/generic/map.h

- › `TSet<T>` – наследник `std::set<T>`
- › `TMap<T>` – наследник `std::map<T>`

Примитивы C++ (строки)

util/generic/strbuf.h

- › **TStringBuf** – аналог `std::string_view`

util/generic/string.h

- › **TString** – CoW строка `char`
- › **TUtf16String** - CoW строка `wchar16`

util/string/cast.h

- › **TString ToString<T>** – эквивалент `std::to_string`
- › **T FromString<T>** – эквивалент `std::unordered_set`
- › `bool TryFromString<T>(..., T* value)`

Примитивы C++ (исключения и ассерты)

Исключения

- › `yexception` – наследник `std::exception`
- › `ythrow` – обертка над `throw`, добавляющая информацию о месте бросания исключений
- › `TString CurrentExceptionMessage()` – текстовое описание исключения

Ассерты и проверки

- › `Y_ASSERT()`
- › `Y_VERIFY()`

Примитивы C++ (время и таймеры)

`util/system/hp_timer.h`

- › `THRTimer` – таймер высокого разрешения


Исключения в CatBoost коде

`TCatBoostException` – базовое исключение, в отличии от `uexception` хранит в себе стектрейс


`CB_ENSURE` – аналог `Y_ENSURE`, бросающий исключение `TCatBoostException`

Code style

Общий стиль кода на C++:

 https://github.com/catboost/catboost/blob/master/CPP_STYLE_GUIDE.md

Расширение стиля для catboost:

 https://github.com/catboost/catboost/blob/master/catboost_command_style_guide_extension.md

Python

 **PEP8** 😊

Особенности сборки C++

Сборка – статическая линковка с библиотеками. Минимум внешних зависимостей позволяет работать на большинстве платформ и дистрибутивов

Система сборки – уmake, часть утилиты уа. Цели сборки описываются в уа.make файлах

Везде, кроме папок util/ и кода cuda ядер разрешен C++17

ya.make

```
LIBRARY() # пример статической библиотеки
```

```
SRCS( # этот макрос содержит список единиц сборки
```

```
visitor.cpp
```

```
...
```

```
GLOBAL cb_dsv_loader.cpp # этот файл будет слинкован в конечную цель сборки (например, нужно для классов-регистраторов)
```

```
)
```

```
PEERDIR( # список зависимостей в виде путей относительно корня репозитория
```

```
    library/dbg_output
```

```
...
```

```
    catboost/libs/quantization
```

```
    catboost/libs/quantization_schema
```

```
)
```

```
# макрос, включающий генератор сериализаторов/десериализаторов для enum и enum class
```

```
GENERATE_ENUM_SERIALIZATION(visitor.h)
```

```
END()
```

Компиляция и запуск тестов

Команда сборки бинарного таргета:

```
./ya make {-r|-d} <target path>
```

Для сборки с системным питоном добавьте ключи

```
-DUSE_ARCADIA_PYTHON=no -DOS_SDK=local -  
DPYTHON_CONFIG=python3-config
```

Чтобы запустить тесты, нужно указать в ya make ключи -t (включает тестирование) -A (запускает все тесты)

```
./ya make -r catboost/pytest python-package/ut
```

ya ide

Для удобства разработки можно сгенерировать проект для JetBrains CLion или QtCreator с помощью команд `./ya ide clion` или `./ya ide qt` соответственно

Demo



Решаем задачу #4 из open_problems.md

Если `learning_rate == 0`, то CatBoost должен
бросать `TCatBoostException`. Добавить в валидацию опций.

Обновляем репозиторий и создаем ветку

```
kirillovs-osx2:catboost kirillovs$ git pull origin master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Total 6 (delta 5), reused 6 (delta 5), pack-reused 0
Unpacking objects: 100% (6/6), done.
From github.com:catboost/catboost
 * branch                master      -> FETCH_HEAD
   607193a5c..ca3d15cd4  master      -> origin/master
Updating 607193a5c..ca3d15cd4
Fast-forward
 catboost/cuda/targets/query_cross_entropy.h | 5 +++++
 1 file changed, 5 insertions(+)
kirillovs-osx2:catboost kirillovs$ git push my_fork master
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 767 bytes | 767.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To github.com:kizill/catboost.git
   607193a5c..ca3d15cd4  master -> master
kirillovs-osx2:catboost kirillovs$ git checkout -b check_zero_learning_rate
Switched to a new branch 'check_zero_learning_rate'
kirillovs-osx2:catboost kirillovs$ █
```

Парсинг опций

Библиотека опций: `catboost/libs/options`

`TCatBoostOptions` – главный класс опций обучения

`TOption<OptionType>` – обертка на единичной опцией или группой опций

Парсинг опций

Библиотека опций: `catboost/libs/options`

`TCatBoostOptions` – главный класс опций обучения

`TOption<OptionType>` – обертка на единичной опцией или группой опций

Парсинг опций

```
namespace NCatboostOptions {
    struct TBoostingOptions {
        explicit TBoostingOptions(ETaskType taskType);

        void Save(NJson::TJsonValue* options) const;
        void Load(const NJson::TJsonValue& options);

        bool operator==(const TBoostingOptions& rhs) const;
        bool operator!=(const TBoostingOptions& rhs) const;

        void Validate() const;

        TOption<float> LearningRate;
        TOption<float> FoldLenMultiplier;
        TOption<ui32> PermutationBlockSize;
        TOption<ui32> IterationCount;
        TOption<ui32> PermutationCount;
        TOption<TOverfittingDetectorOptions> OverfittingDetector;
        TOption<EBoostingType> BoostingType;
        TCpuOnlyOption<bool> ApproxOnFullHistory;

        TGPUOnlyOption<ui32> MinFoldSize;
        TGPUOnlyOption<EDataPartitionType> DataPartitionType;
    };
}
```

Вносим правки

```
kirillovs-osx2:catboost kirillovs$ git diff
diff --git a/catboost/libs/options/boosting_options.cpp b/catboost/libs/options/boosting_options.cpp
index b748dcd2a..ce7204c3b 100644
--- a/catboost/libs/options/boosting_options.cpp
+++ b/catboost/libs/options/boosting_options.cpp
@@ -4,6 +4,8 @@
#include <catboost/libs/logging/logging.h>
#include <catboost/libs/logging/logging_level.h>

+#include <util/generic/ymath.h>
+
NCatboostOptions::TBoostingOptions::TBoostingOptions(ETaskType taskType)
    : LearningRate("learning_rate", 0.03)
    , FoldLenMultiplier("fold_len_multiplier", 2.0)
@@ -60,7 +62,11 @@ void NCatboostOptions::TBoostingOptions::Validate() const {
}

CB_ENSURE(!(ApproxOnFullHistory.GetUnchecked() && BoostingType.Get() == EBoostingType::Plain), "Can't use approx-on-full-history with Plain boosting-type");
- if (LearningRate.IsSet() && LearningRate.Get() > 1) {
-     CATBOOST_WARNING_LOG << "learning rate is greater than 1. You probably need to decrease learning rate." << Endl;
+ if (LearningRate.IsSet()) {
+     CB_ENSURE(Abs(LearningRate.Get()) > std::numeric_limits<float>::epsilon(), "Learning rate should be non-zero");
+     if (LearningRate.Get() > 1) {
+         CATBOOST_WARNING_LOG
+         << "learning rate is greater than 1. You probably need to decrease learning rate." << Endl;
+     }
+ }
}
```


Добавляем CLI тест

```
3917
3918 def test_zero_learning_rate():
3919     train_path = yatest.common.test_output_path('train')
3920     cd_path = yatest.common.test_output_path('train.cd')
3921
3922     open(cd_path, 'wt').write(
3923         '0\tNum\n'
3924         '1\tNum\n'
3925         '2\tTarget\n')
3926     np.savetxt(train_path, [
3927         [0, 1, 2],
3928         [1, 1, 1]], delimiter='\t', fmt='%.4f')
3929     cmd = (CATBOOST_PATH, 'fit',
3930         '-f', train_path,
3931         '--cd', cd_path,
3932         '--learning-rate', '0.0',
3933         )
3934     with pytest.raises(yatest.common.ExecutionError):
3935         yatest.common.execute(cmd)
3936
```


Добавляем python-package тест

```
701
702 def test_zero_learning_rate(task_type):
703     train_pool = Pool(TRAIN_FILE, column_description=CD_FILE)
704     model = CatBoost({
705         'learning_rate': 0.0,
706         'loss_function': 'RMSE',
707         'task_type': task_type,
708         'devices': '0'})
709     with pytest.raises(CatboostError, message='Learning rate should be non-zero'):
710         model.fit(train_pool)
```

Проверяем новые тесты

```
kirillovs-osx2:catboost kirillovs$ yar -tA catboost/pytest catboost/python-package/ut/medium/ -F test.py::test_zero_learning_rate*
|86.2%| [TS] $(B)/catboost/python-package/ut/medium/test-results/catboost-python-package-ut-medium/{meta.json ... results_accumulator
|86.2%| [TS] $(B)/catboost/python-package/ut/medium/test-results/catboost-python-package-ut-medium/{meta.json ... results_accumulator
|86.2%| [TS] $(B)/catboost/python-package/ut/medium/test-results/catboost-python-package-ut-medium/{meta.json ... results_accumulator
----- [TS] $(B)/result.log
Number of suites skipped by name: 4, by filter test.py::test_zero_learning_rate*

Total 2 suites:
      2 - GOOD
Total 2 tests:
      2 - GOOD
Ok
```

Коммитим и пушим форкнутый репозиторий

```
kirillovs-osx2:catboost kirillovs$ git branch
0.11.2
* check_zero_learning_rate
make_ya_ide_work
master
v0.10.0
v0.8.1.1
v0.9
v0.9.1.1
kirillovs-osx2:catboost kirillovs$ git commit -a -m "add zero learning rate check and tests"
[check_zero_learning_rate 76d2e662e] add zero learning rate check and tests
3 files changed, 39 insertions(+), 2 deletions(-)
kirillovs-osx2:catboost kirillovs$ git push my_fork check_zero_learning_rate
Counting objects: 12, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 1.73 KiB | 886.00 KiB/s, done.
Total 12 (delta 10), reused 0 (delta 0)
remote: Resolving deltas: 100% (10/10), completed with 10 local objects.
remote:
remote: Create a pull request for 'check_zero_learning_rate' on GitHub by visiting:
remote:      https://github.com/kizill/catboost/pull/new/check_zero_learning_rate
remote:
To github.com:kizill/catboost.git
* [new branch]      check_zero_learning_rate -> check_zero_learning_rate
```

Публикуем pull request

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base repository: catboost/catboost

base: master

head repository: kizill/catboost

compare: check_zero_learning_rate

✓ Able to merge.

These branches can be automatically merged.

add zero learning rate check and tests

Write

Preview

Before submitting a pull request, please do the following steps:

1. Read instructions for contributors [here]
(<https://tech.yandex.com/catboost/doc/dg/concepts/development-and-contributions-docpage/>).

2. Run `ya make` in catboost folder to make sure the code builds.

3. Add tests that test your change.

4. Run tests using `ya make -t -A` command.

5. If you haven't already, complete the CLA.

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

☒ Allow edits from maintainers.

[Learn more](#)

Create pull request

Reviewers

Suggestions

andrey-khropov

Request

dbakshee

Request

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Helpful resources

[Contributing](#)

36

Удачи!

Станислав Кириллов

Ведущий разработчик CatBoost