

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**АЛГОРИТМЫ НАСТРОЙКИ ГИПЕРПАРАМЕТРОВ НА ОСНОВЕ
ОБЪЕДИНЕНИЯ АПРИОРНЫХ И АПОСТЕРИОРНЫХ ЗНАНИЙ О
ЗАДАЧЕ**

Автор: Смирнова Валентина Сергеевна _____

Направление подготовки: 01.03.02 Прикладная
математика и информатика

Квалификация: Бакалавр

Руководитель: Фильченков А.А., к.ф.-м.н. _____

К защите допустить

Руководитель ОП Парфенов В.Г., проф., д.т.н. _____

« ____ » _____ 20__ г.

Санкт-Петербург, 2020 г.

Обучающийся Смирнова В.С.

Группа М3435 Факультет ИТиП

Направленность (профиль), специализация

Математические модели и алгоритмы в разработке программного обеспечения

ВКР принята « ____ » _____ 20 ____ г.

Оригинальность ВКР ____ %

ВКР выполнена с оценкой _____

Дата защиты « ____ » _____ 20 ____ г.

Секретарь ГЭК Павлова О.Н. _____

Листов хранения _____

Демонстрационных материалов/Чертежей хранения _____

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

УТВЕРЖДАЮ

Руководитель ОП
проф., д.т.н. Парфенов В.Г. _____
« ____ » _____ 20__ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Обучающийся Смирнова В.С.

Группа М3435 **Факультет/институт/кластер** ИТиП

Квалификация Бакалавр

Направление подготовки 01.03.02 Прикладная математика и информатика

Направленность (профиль) образовательной программы Математические модели и алгоритмы в разработке программного обеспечения

Специализация

Тема ВКР Алгоритмы настройки гиперпараметров на основе объединения априорных и апостериорных знаний о задаче

Руководитель Фильченков А.А., к.ф.-м.н, доцент факультета информационных технологий и программирования, Университет ИТМО

2 Срок сдачи студентом законченной работы до « ____ » _____ 20__ г.

3 Техническое задание и исходные данные к работе

Требуется разработать алгоритм настройки гиперпараметров на основе объединения априорных и апостериорных знаний о задаче

4 Содержание выпускной работы (перечень подлежащих разработке вопросов)

В работе должна быть показана эффективность разработанного решения по сравнению с существующими

5 Перечень графического материала (с указанием обязательного материала)

Графические материалы и чертежи работой не предусмотрены

6 Исходные материалы и пособия

- а) Hutter, F., Hoos, H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. // LION'11. — 2011
- б) Efficient and robust automated machine learning / M. Feurer // Advances in neural information processing systems. — 2015.
- в) Leite R., Brazdil P. Active Testing Strategy to Predict the Best Classification Algorithm via Sampling and Metalearning. // ECAI. — 2010.

7 Дата выдачи задания « ____ » _____ 20__ г.

Руководитель ВКР _____

Задание принял к исполнению _____

« ____ » _____ 20__ г.

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Обучающийся Смирнова Валентина Сергеевна

Наименование темы ВКР Алгоритмы настройки гиперпараметров на основе объединения априорных и апостериорных знаний о задаче

Наименование организации, где выполнена ВКР: Национальный Исследовательский Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Разработать алгоритм настройки гиперпараметров на основе объединения априорных и апостериорных знаний о задаче

2 Задачи, решаемые в ВКР:

- а) изучить существующие решения поставленной задачи;
- б) предложить и реализовать новый алгоритм;
- в) провести эксперименты, показывающие эффективность решения.

3 Число источников, использованных при составлении обзора: 0

4 Полное число источников, использованных в работе: 7

5 В том числе источников по годам:

Отечественных			Иностранных		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
0	0	0	5	2	0

6 Использование информационных ресурсов Internet: нет

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	Раздел работы
Пакет numpy	Не предусмотрено
Пакет pandas	Не предусмотрено
Пакет robo	Не предусмотрено
Пакет matplotlib	Не предусмотрено
Пакет george	Не предусмотрено

8 Краткая характеристика полученных результатов

Был предложен и реализован эффективный алгоритм, решающий поставленную задачу.

9 Гранты, полученные при выполнении работы

Отсутствуют

10 Наличие публикаций и выступлений на конференциях по теме работы

а) IX Конгресс Молодых Учёных

Обучающийся Смирнова В.С. _____

Руководитель ВКР Фильченков А.А. _____

« _____ » _____ 20__ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. Обзор существующих решений	7
1.1. Определения и ключевые понятия	7
1.2. Обзор классификаторов	7
1.3. Меры оценки качества классификации	9
1.4. Обзор подходов к оптимизации гиперпараметров	10
1.4.1. Байесовская оптимизация	12
Выводы по главе 1	14
2. Предложенное решение	15
2.1. Параметры для оптимизации	15
2.1.1. Модель оптимизации	15
2.1.2. Классификатор	15
2.1.3. Целевая функция	17
2.2. Метрика для определения подобия задач	18
2.3. Расширение Байесовской оптимизации	18
2.4. Данные	18
Выводы по главе 2	18
3. Анализ полученных результатов	19
Выводы по главе 3	19
ЗАКЛЮЧЕНИЕ	20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	21
ПРИЛОЖЕНИЕ А. Результаты классической Байесовской оптимизации ..	22

ВВЕДЕНИЕ

Задача классификации – построить алгоритм (классификатор), который по набору признаков вернул бы метку класса или вектор оценок принадлежности (апостериорных вероятностей) к каждому из классов. Основная её цель – максимально точно определить метку класса для заданного объекта. Задача широко применяется во многих областях:

- Медицинская диагностика: по набору медицинских характеристик требуется поставить диагноз
- Геологоразведка: по данным зондирования почв определить наличие полезных ископаемых
- Оптическое распознавание текстов: по отсканированному изображению текста определить цепочку символов, его формирующих
- Кредитный скоринг: по анкете заемщика принять решение о выдаче/отказе кредита
- Синтез химических соединений: по параметрам химических элементов спрогнозировать свойства получаемого соединения

Найти и обучить эффективный алгоритм классификации – трудоёмкая задача, которая включает в себя выбор самого классификатора, сбор и разметку данных (в случае обучения с учителем), и непосредственно настройку гиперпараметров классификатора. Так как гиперпараметры задаются до начала обучения и не изменяются в его ходе, а при этом могут существенно влиять на результат обучения, то появляется задача оптимизации гиперпараметров.

Оптимизация гиперпараметров — задача машинного обучения по выбору набора оптимальных гиперпараметров для обучающего алгоритма. Одни и те же виды моделей машинного обучения могут требовать различные предположения, веса или скорости обучения для различных видов данных. Эти параметры называются гиперпараметрами и их следует настраивать так, чтобы модель могла оптимально решить задачу обучения. Для этого находится кортеж гиперпараметров, который даёт оптимальную модель, оптимизирующую заданную функцию потерь на заданных независимых данных. Такая задача несёт название AutoML [4], основная её задача – сделать процесс машинного обучения доступным не только для экспертов в области ML, но и для любого пользователя.

Интерес к задаче AutoML растёт, проводятся конференции и соревнования по её решению. В частности, каждые два года проходит соревнование AutoML Challenge. В августе 2019 года происходило мероприятие, посвящённое этой теме – The Third International Workshop on Automation in Machine Learning. Согласно выпуску Forbes за декабрь 2018 года, это один из пяти трендов в развитии машинного обучения в 2019 году. Тема AutoML появляется все чаще и чаще в дискуссиях и публикациях. Решения этой задачи уже используются в автономных машинах, предсказании цен и многих других областях.

Существующие решения [2, 3, 5, 6] для задачи оптимизации гиперпараметров основываются на случайной расстановке гиперпараметров и дальнейшей их настройке. В данной работе будет предложен подход, основанный не на случайной расстановке первичных гиперпараметров, а на особом подходе, основанном на результатах обучения смежных задач.

ГЛАВА 1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Более подробно рассмотрим задачу классификации, алгоритмы ее решения и способы оценки качества полученной классификации. Также опишем понятие модели алгоритма классификации и основные методы настройки её гиперпараметров, применимые ко многим алгоритмам машинного обучения. Заметим, что на сегодняшний момент не предложено способов не случайной расстановки гиперпараметров и дальнейшей их настройки, на основе результатов обучения на схожих задачах.

1.1. Определения и ключевые понятия

Для начала введем определения и ключевые понятия, которые будут использоваться в дальнейшей работе.

- классификатор – параметризованный алгоритм, решающий задачу классификации
- конфигурация – фиксированный набор параметров классификатора
- алгоритм – пара из классификатора и его конфигурации
- решаемая задача – алгоритм с настроенными гиперпараметрами на конкретном датасете
- решённая задача – алгоритм с оптимизированными гиперпараметрами на конкретном датасете
- соседняя задача – ближайшая задача к решаемой
- текущее решение (текущая задача) – решаемая задача, для которой могут использоваться сведения из решённых (соседних) задач

1.2. Обзор классификаторов

В данной части рассмотрим популярные модели для решения задачи классификации. Вспомним, что цель задачи классификации – наиболее точно определять метку класса по заданному объекту.

Итак, наиболее используемые на сегодняшний момент модели классификаторов:

Линейная регрессия Можно представить в виде уравнения, которое описывает прямую, наиболее точно показывающую взаимосвязь между входными переменными X и выходными переменными Y

Логистическая регрессия По аналогии с линейной регрессией требуется найти коэффициенты для входных данных, но уже с помощью нелинейной или логистической функции.

Линейный дискриминантный анализ (LDA) Состоит из статистических свойств данных, рассчитанных для каждого класса

- Среднее значение для каждого класса
- Дисперсия, рассчитанная по всем классам

Деревья принятия решений Представима в виде бинарного дерева, где каждый узел представляет собой входную переменную и точку разделения для этой переменной (при условии, что переменная — число)

Наивный Байесовский классификатор Состоит из двух типов вероятностей, которые рассчитываются с помощью тренировочных данных:

- а) Вероятность каждого класса
- б) Условная вероятность для каждого класса при каждом значении x

К-ближайших соседей (KNN) Предсказание метки класса делается на основе меток k ближайших соседей

Метод опорных векторов (SVM) Суть метода заключается в построении гиперплоскости, разделяющей классы

Бэггинг и случайный лес (RandomForest) Один из наиболее эффективных алгоритмов классификации, берётся множество подвыборок из данных, считается среднее значение для каждой, а затем усредняются результаты для получения лучшей оценки действительного среднего значения

Бустинг и AdaBoost : принадлежит семейству ансамблевых алгоритмов, суть которых заключается в создании сильного классификатора на основе нескольких слабых

Многослойный персептрон (Multilayered perceptron) Класс искусственных нейронных сетей прямого распространения, состоящих как минимум из трех слоёв: входного, скрытого и выходного

В работе мы заострим внимание на модели случайного леса как самой эффективной на сегодняшний момент и модели многослойного персептрона, так как он обладает наибольшим количеством гиперпараметров и также является достаточно эффективным.

1.3. Меры оценки качества классификации

Кроме выбора алгоритма классификации и его гиперпараметров, обучить модель, нужно ещё каким-то образом оценить качество работы обученного алгоритма. Для этого датасет делится на 2 части: *train* (на которой модель обучается) и *test* или *validate* (на которой оценивается качество классификации). В данной части мы рассмотрим существующие меры оценки качества классификации и выделим существенные для нашей задачи. Для этого сначала вспомним базовые понятия:

- Верно-положительными (TP) называются объекты, которые были классифицированы как положительные и действительно являются таковыми
- Верно-отрицательными (TN) называются объекты, которые были классифицированы как отрицательные и действительно таковые
- Ложно-положительными (FP) называются объекты, которые были классифицированы как положительные, но фактически отрицательные
- Ложно-отрицательными (FN) называются объекты, которые были классифицированы как отрицательные, но фактически положительные

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Одна из наиболее простых и популярных мер оценки качества — *F-мера* или *F-score*. Считается она следующим образом:

$$F - score = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

Приемущество данной меры в том, что её достаточно просто считать и результат отлично подходит для целевой функции оптимизации, о которой мы поговорим в следующей части.

Также существует *кривая ошибки* или *ROC-curve (Receiver Operating Characteristic)*. Суть данной меры состоит в том, что считается площадь кривой под графиком уровня верно-положительных предсказанных экземпляров от уровня ложно-положительных. Не будем заострять на ней внимание, так как к нашей задаче она не подходит из-за трудоёмкости расчёта.

1.4. Обзор подходов к оптимизации гиперпараметров

Существует несколько подходов к задаче оптимизации гиперпараметров. В данной части рассмотрим наиболее популярные из них, оценим преимущества и недостатки.

Поиск по решётке По сути, данный алгоритм делает полный перебор всех возможных моделей и конфигураций.

Доступные реализации

- LIBSVM
- scikit-learn
- Talos

Достоинства Гарантирована будет найдена наилучшая конфигурация.

Недостатки Слишком большие затраты на обучение.

Случайный поиск Отличается от поиска по решётке тем, что идёт не полный перебор всех конфигураций, а случайная их выборка.

Доступные реализации

- hyperopt
- scikit-learn

- H2O AutoML
- Talos

Достоинства Меньшие затраты на обучение. Существует вероятность нахождения наилучшей конфигурации за наименьшее время.

Недостатки Неопределённое количество времени на поиск наилучшей конфигурации.

Байесовская оптимизация Метод, основанный на обращении к функции «чёрного ящика» с шумом. Для задачи оптимизации гиперпараметров строит стохастическую модель из отображения из конфигурации в целевую функцию, применённую на валидационном наборе данных.

Доступные реализации

- Spearmint
- Bayesopt
- MOE
- Auto-WEKA
- Auto-sklearn
- mlrMBO
- tuneRanger
- BOCS
- SMAC

Достоинства Небольшие затраты на обучение, количество итераций задаётся вручную, адаптируется под значимость каждого гиперпараметра для конкретной задачи.

Недостатки Сложность реализации и использования.

Оптимизация на основе градиентов Для определённых алгоритмов обучения вычисляется градиент гиперпараметров и оптимизируется с помощью градиентного спуска.

Доступные реализации

— hypergrad

Достоинства Небольшие затраты на обучение, неплохой результат.

Недостатки Сложность понимания и реализации, небольшое количество доступных реализаций.

Эволюционная оптимизация Также, как и Байесовская оптимизация, основывается на обращениях к функции «чёрного ящика» с шумом, однако для поиска гиперпараметров для заданного алгоритма использует эволюционные подходы (алгоритмы)[1].

Доступные реализации

— TROT
— devol
— deap

Достоинства Небольшие затраты на обучение, неплохой результат.

Недостатки Используется только для статистических алгоритмов. Давайте немного подробнее рассмотрим Байесовскую оптимизацию.

1.4.1. Байесовская оптимизация

Как уже говорилось в предыдущей части 1.4, Байесовская оптимизация основывается на обращении к функции «чёрного ящика» с шумом. Кроме того, подход требует определить модель, на которой будет основываться сама Байесовская оптимизация, максимизатор, способ задания первичных гиперпараметров и непосредственно целевую функцию. Сам алгоритм представляет из себя следующую цепочку действий:

Точка в алгоритме определяется конфигурацией гиперпараметров. Лучшее значение — минимальное значение, выданное целевой функцией в данной точке. Наиболее интересная и значимая часть в алгоритме 1 — выбор

Листинг 1 – Байесовская оптимизация

```

задать первичные гиперпараметры
for количество итераций do
    выбрать следующую точку для рассмотрения
    получить результат целевой функции в этой точке
    сохранить лучшее значение и конфигурацию
end for

```

следующей для рассмотрения точки. Именно в этой части модель обучается на уже рассмотренных точках и обновляет так называемую *функцию выгоды* (*acquisition function*), после чего в максимуме функции выгоды будет место с наибольшей неопределённостью, значит, там нав и необходимо будет «смотреть» на точку. Подробнее про подход описано в AutoML Book[[automlbook19a](#)].

Модель Определяет, как будет обновляться пространство гиперпараметров модели оптимизатора. Важно отметить, что гиперпараметры оптимизатора никаким образом не связаны с гиперпараметрами классификатора (которые мы пытаемся оптимизировать). Наиболее распространённые модели:

- а) GaussianProcess
- б) GaussianProcessMCMC
- в) RandomForest
- г) WrapperBohamiann
- д) DNGO

У каждой модели есть свои преимущества и свои недостатки, например, в Гауссовском процессе нет возможности работать с категориальными признаками, DNGO же обладает наименее точным результатом.

Максимизатор В Байесовской оптимизации работает с функцией выгоды, которая находит следующую наиболее интересную точку. Максимизаторы бывают:

- а) RandomSampling
- б) SciPyOptimizer
- в) DifferentialEvolution

Функция выгоды Служит для определения наименее информативной области. В этой области меньше всего информации о значении функции, которую мы пытаемся аппроксимировать, поэтому в экстремуме функции выгоды будет находиться наиболее интересная на данный момент конфигурация гиперпараметров. Популярные функции выгоды:

- а) EI
- б) LogEI
- в) PI
- г) LCB

Целевая функция В общем случае принимает на вход конфигурацию гиперпараметров и отдаёт значение той самой функции «чёрного ящика», значение которой минимизируется в процессе Байесовской оптимизации. В случае рассматриваемой нами задачи классификации, целевая функция будет обучать классификатор с полученными гиперпараметрами на третировочной выборке и вернёт обратное значение (так как функция минимизируется) меры оценки качества классификации, посчитанной на валидационной выборке.

Выводы по главе 1

В первой главе рассмотрены ключевые определения и понятия, необходимые для понимания решаемой задачи. Оценены преимущества и недостатки параметров для оптимизации гиперпараметров, такие как классификаторы, меры оценки качества классификации, подходы к задаче оптимизации и их параметры. Также подробно рассмотрен один из основных и наиболее популярных подходов – Байесовская оптимизация. Описаны функции и задачи модели оптимизатора, его целевой функции, функции выгоды и максимизатора.

ГЛАВА 2. ПРЕДЛОЖЕННОЕ РЕШЕНИЕ

За основу предложенного решения была выбрана Байесовская оптимизация, а именно реализация *RoBO (Robust Bayesian Optimization Framework)*[7]. Для однозначного понимания, далее будем называть её «классической Байесовской оптимизацией», чтобы иметь возможность отличать от непосредственно предложенного решения. Задача данного исследования – добиться лучших результатов оптимизации. Под «лучшими» результатами стоит понимать лучшее (минимальное) значение *incubment*¹, полученное на более ранней итерации оптимизации. Поэтому достаточно задать фиксированные параметры классического решения и сравнивать его результаты с результатами доработанного решения с теми же параметрами. Далее будут рассмотрены выбранные в работе параметры для классической Байесовской оптимизации.

2.1. Параметры для оптимизации

В первой главе 1.4.1 сравнили параметры, необходимые для запуска классической Байесовской оптимизации. В текущей главе будут выбраны наиболее подходящие нам параметры.

2.1.1. Модель оптимизации

В качестве модели оптимизации в работе используется Гауссовский процесс. Как известно, оптимизаторы, построенные на модели случайного леса пользуются бóльшим спросом по причине того, что даёт лучшие результаты, однако, как очевидно из названия, данная модель основана на случайности, что не несёт под собой точного математического обоснования, в отличие от Гауссовского процесса. Именно по этой причине в работе используется именно Гауссовский процесс.

2.1.2. Классификатор

2.1.2.1. Обоснование

При выборе классификатора, стоит помнить, что мы решаем задачу оптимизации гиперпараметров, значит, набор гиперпараметров классификатора должен соответствовать следующим свойствам:

- а) иметь ненулевой конечный набор гиперпараметров
- б) настраиваемые гиперпараметры должны быть вещественными числами

¹значение целевой функции

- в) не иметь категориальных гиперпараметров, так как в качестве модели оптимизатора выступает Гауссовский процесс, который не предусматривает работу с категориальными гиперпараметрами

К сожалению, не существует модели, удовлетворяющей всем перечисленным свойствам, поэтому было принято решение игнорировать (фиксировать определённое значение и не изменять в процессе всей оптимизации) категориальные гиперпараметры (представлять их в виде чисел было бы некорректно) и приводить числа с плавающей точкой к целым на местах гиперпараметров, где требуются целочисленные значения (данное пренебрежение корректно, так как значения гиперпараметров, требующие целочисленности, располагаются в диапазоне, много большем, чем потеря при округлении).

Кроме того, стоит помнить, что необходимо подобрать классификатор с как можно большим количеством подходящих гиперпараметров, чтобы была возможность корректно оценить работу оптимизатора. Более подробно: гиперпараметры бывают более значимы для классификатора или менее значимы, их значимость определяет именно оптимизатор в ходе оптимизации. Поэтому в выбранном классификаторе должны присутствовать и те, и другие.

Проанализировав все требования к классификатору, был выбран многослойный парцептрон Румельхарта (Рисунок 1).

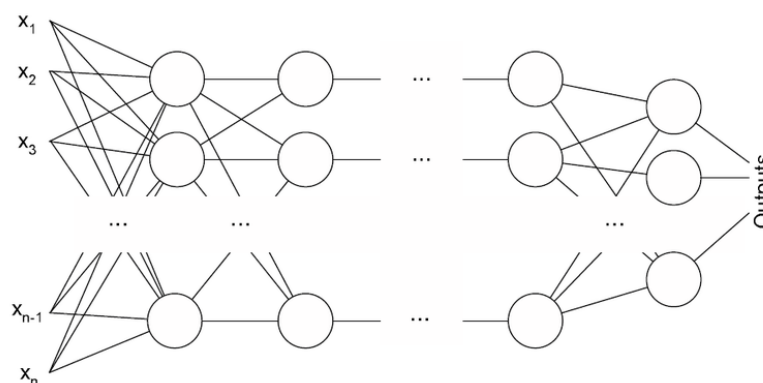


Рисунок 1 – Многослойный парцептрон Румельхарта

2.1.2.2. Набор гиперпараметров

Гиперпараметры многослойного парцептрона Румельхарта, которые будут использованы для оптимизации (в круглых скобках указан тип гиперпараметра, в квадратных – диапазон его значений):

- hidden layer sizes (*int*) – число скрытых слоёв [1, 150]

- α (*float*) – L2 штраф [0.00001, 0.01]
- learning rate initial (*float*) – начальная скорость обучения [0.0001, 0.1]
- max iterations (*int*) – максимальное число итераций [50, 300]
- validation fraction (*float*) – доля данных обучения, отведенных в качестве проверки для преждевременной остановки [0.01, 0.9]
- β_1 (*float*) – скорость экспоненциального убывания для оценок вектора первого момента [0.09, 0.9]
- β_2 (*float*) – Скорость экспоненциального убывания для оценок вектора второго момента [0.0999, 0.999]
- n iterations no change (*int*) – максимальное число эпох, пройденных без прогресса [5, 15]

2.1.3. Целевая функция

Как было упомянуто в части 1.4.1, целевая функция в нашем случае должна отдавать значение меры качества классификации, вернее обратное её значение, так как идёт процесс минимизации. Поэтому вернёмся к части 1.3 и выберем подходящую нам меру оценки качества классификации.

Мера оценки качества классификации Наиболее подходящей мерой в нашем случае является *F-score*, так как она проста в вычислении и от неё легко берётся обратное значение: $(1 - F\text{-score})$ – именно это значение будет выдаваться целевой функцией при каждом обращении к ней. Более детально рассмотрим процесс, происходящий внутри целевой функции. До запуска оптимизации, функция инициализируется выборкой, разделённой на 2 части: *train* и *validate*. Далее запускается оптимизационный процесс, где вызывается наша целевая функция. При каждом обращении, на вход функции подаётся конфигурация гиперпараметров, сгенерированная алгоритмом оптимизации для классификатора. После чего запускается процесс обучения классификатора с заданной конфигурацией на тренировочной части выборки, далее запускается процесс предсказания меток классов на валидационной части выборки, считается значение *F-score* и возвращается описанное ранее значение целевой функции. Затем оптимизатор анализирует полученный результат, перестраивает конфигурацию и повторяет до тех пор, пока не достигнет последней итерации.

2.2. Метрика для определения подобия задач

оаоа

2.3. Расширение Байесовской оптимизации

аоаоао

2.4. Данные

Выводы по главе 2

аоаооа

ГЛАВА 3. АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

аооаоа

Выводы по главе 3

аоаооа

ЗАКЛЮЧЕНИЕ

В данном разделе размещается заключение. А

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Algorithms for Hyper-Parameter Optimization / J. S. Bergstra [et al.] // Advances in Neural Information Processing Systems 24 / ed. by J. Shawe-Taylor [et al.]. — Curran Associates, Inc., 2011. — P. 2546–2554. — URL: <http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf>.
- 2 Efficient and Robust Automated Machine Learning / M. Feurer [et al.] // Advances in Neural Information Processing Systems 28 / ed. by C. Cortes [et al.]. — Curran Associates, Inc., 2015. — P. 2962–2970. — URL: <http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf>.
- 3 *Falkner S., Klein A., Hutter F.* BOHB: Robust and Efficient Hyperparameter Optimization at Scale // Proceedings of the 35th International Conference on Machine Learning (ICML 2018). — 07/2018. — P. 1436–1445.
- 4 *Feurer M., Hutter F.* Hyperparameter Optimization // AutoML: Methods, Sytems, Challenges / ed. by F. Hutter, L. Kotthoff, J. Vanschoren. — Springer, 05/2019. — Chap. 1. P. 3–33.
- 5 *Hutter F., Hoos H. H., Leyton-Brown K.* Sequential Model-Based Optimization for General Algorithm Configuration (extended version) : tech. rep. / University of British Columbia, Department of Computer Science. — 2010. — TR-2010–10. — Available online: <http://www.cs.ubc.ca/~hutter/papers/10-TR-SMAC.pdf>.
- 6 *Lindauer M., Hutter F.* Warmstarting of Model-based Algorithm Configuration. — 2017. — arXiv: 1709.04636 [cs.AI].
- 7 RoBO: A Flexible and Robust Bayesian Optimization Framework in Python / A. Klein [et al.] // NIPS 2017 Bayesian Optimization Workshop. — 12/2017.

ПРИЛОЖЕНИЕ А. РЕЗУЛЬТАТЫ КЛАССИЧЕСКОЙ БАЙЕСОВСКОЙ ОПТИМИЗАЦИИ

Имя датасета	Лучшее значение (1 - F-score)	Номер итерации
page-blocks	0.1572831335432724	42
robot-failures-lp1	0.14102564102564108	33
mfeat-fourier	0.1475854394148598	2
jungle-chess-2pcs-raw-endgame-complete	0.2719523672987666	27
heart-switzerland	0.6858119658119658	91
gas-drift-different-concentrations	0.01891421038359764	77
wall-robot-navigation	0.01529599728237685	57
jungle-chess-2pcs-endgame-panther-elephant	0.0	88
leaf	0.36017355660212813	73
PopularKids	0.06352989828599576	40
mfeat-karhunen	0.014079861405182248	20
diggle-table-a2	0.9743589743589743	84
rmftsa-sleepdata	0.9278268809884568	27
semeion	0.05376620700232804	28
desharnais	0.2667332667332668	86
teachingAssistant	0.4071969696969696	44
collins	0.8884880528144286	38
volcanoes-a3	0.7689402697495183	61
artificial-characters	0.3613704595813323	30
volcanoes-a4	0.7972789115646258	51
glass	0.5231009070294785	68
nursery	0.04052981019439761	61
shuttle	0.0	30
segment	0.02531199490574576	62
heart-long-beach	0.652975912975913	91
vertebra-column	0.20623124372223145	80

cnae-9	0.034818671429924564	26
jannis	0.9999969017034435	10
wine-quality-white	0.750911817087637	40
vehicle	0.34988621629488503	36
ecoli	0.24709595959595954	58
eye-movements	0.5769399013962551	16
seeds	0.11004901960784308	96
car	0.13424741140312846	74
fabert	0.8754012841091493	0
breast-tissue	0.6674242424242425	64
thyroid-allbp	0.5548263021310648	81
gas-drift	0.011142829995818726	81
mfeat-factors	0.08802270506646293	53
volcanoes-d1	0.7974875207986689	10
har	0.013956849985395814	39
satimage	0.10895145985989307	73
Fashion-MNIST	0.1732567305672541	47
seismic-bumps	0.03781821523757012	34
pokerhand	0.28102343233242333	84
helena	0.981850107656847	98
thyroid-allhyper	0.5208075903466287	14
wine	0.9846153846153847	99
balance-scale	0.07249605495219524	46
microaggregation2	0.5627566824728257	44
steel-plates-fault	0.7677201034057892	61
tae	0.3234408602150539	22
mfeat-pixel	0.5312663695353331	92
gina-prior2	0.08624343753649555	81
synthetic-control	0.0	7
cmc	0.4217043857598791	12
energy-efficiency	0.8637585814858542	22
iris	0.0	2
fars	0.4350632170299382	22

abalone	0.6296727096012757	95
prnn-viruses	0.0	62
Indian-pines	0.5687291627863662	4
coverttype	0.4420032958440877	5
JapaneseVowels	0.44703798338069056	91
user-knowledge	0.1050103519668737	29
spectrometer	0.9662337662337662	47
hayes-roth	0.12222222222222212	34
robot-failures-lp5	0.2971834250941756	59
prnn-fglass	0.3836808236808237	27
waveform-5000	0.12700439195696356	27
zoo	0.0	12
cardiotocography	0.06492996280208396	96
mfeat-morphological	0.5482421206646882	6
volcanoes-a1	0.8019336485603352	98
tamilnadu-electricity	0.0755314544908181	40
LED-display-domain-7digit	0.21607382867705938	44