

РЕФЕРАТ

Пояснительная записка _ с., _ рис., _ источников, 1 прил.

ПРОГРАММНАЯ ПЛАТФОРМА, ОПТИМИЗАЦИЯ ГИПЕРПАРАМЕТРОВ, ПРОСТРАНСТВО ПОИСКА

Выпускная работа нацелена на разработку программной платформы для оптимизации гиперпараметров. В ходе работы над выпускной работой была сформулирована постановка задачи, изучены основные алгоритмы оптимизации, проведен анализ существующих решений, проектирование программной платформы и ее разработка.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 ОБЗОР ПОДХОДОВ ОПТИМИЗАЦИИ ГИПЕРПАРАМЕТРОВ	6
1.1 Постановка задачи	6

ВВЕДЕНИЕ

В связи с большим количеством программного обеспечения, разрабатываемого на языках высокого уровня становится вопрос об эффективности используемых алгоритмов и их оптимальной настройке.

Множество алгоритмов, в том числе машинного обучения, могут требовать различные предположения, веса, характеристики для их работы. Параметры алгоритмов, значения которых определяются непосредственно перед запуском данных алгоритмов, называются гиперпараметрами.

Обычно слово "гиперпараметр" рассматривается в контексте машинного обучения, в данной же работе данное понятие будет рассмотрено для более общего контекста. В простейшем случае гиперпараметрами могут быть аргументы алгоритмов для вычисления математических функций. Например, для алгоритма вычисляющего значения функции $f(x, y) = \sin(x) + \sin(y)$ гиперпараметрами будут вещественнозначные аргументы x и y .

Гиперпараметры в том числе используются компиляторами (например GCC/Clang) для настройки процесса компиляции и линковки (оптимизационные флаги). В машинном обучении это могут быть веса нейронов в нейронных сетях и ошибка на выходе сети, расстояние между объектами в кластеризации и т.д. В генетических алгоритмах это может быть количество особей в популяции, настройки мутаций и т.д.

Целью данной работы является разработка программной платформы для оптимизации гиперпараметров.

Для достижения поставленной цели необходимо решить следующие задачи:

- Выполнить постановку задачи;
- Изучить основные подходы для решения данной задачи;
- Рассмотреть существующие решения;
- Выполнить проектирование программной платформы, а также произвести её разработку.

Описать разделы ВКР

1 ОБЗОР ПОДХОДОВ ОПТИМИЗАЦИИ ГИПЕРПАРАМЕТРОВ

1.1 Постановка задачи

Пусть A - алгоритм, принимающий N гиперпараметров для своей настройки. Будем обозначать Λ_n - область возможных значений гиперпараметра. Пространством поиска мы назовём $\Lambda = \Lambda_1 \times \Lambda_2 \times \dots \times \Lambda_n$. Конфигурацией мы будем называть $\lambda \in \Lambda$ - вектор значений гиперпараметров, принадлежащий пространству поиска.

Область возможных значений гиперпараметров может быть вещественнозначной (скорость обучения), целочисленной, булевой или категориальной.

Целевой функцией алгоритма A называется вещественнозначная или целочисленная функция, принимающая на вход конфигурацию, подлежащая оптимизации. (можно сказать, что целевая функция возвращает некоторую целевую метрику, например, функция, возвращающая время компиляции некоторой программы и данное время необходимо оптимизировать), ее обозначение: f_A .

В данной терминологии задача ставится следующим образом: необходимо найти:

$$\lambda^* = \operatorname{argmin}_{\lambda \in \Lambda} f_A(\lambda) \quad (1.1)$$

Т.е необходимо найти конфигурацию, при которой целевая функция алгоритма принимает минимальное значение.

Стоит отметить, что пространства поиска могут быть очень большими и полный перебор всевозможных конфигураций будет невыполним.

В данной ситуации используются итеративные методы оптимизации, их суть состоит в следующем:

- 1) Выбирается начальная конфигурация λ_0 .
- 2) Полагается $k = 0$ - счётчик итераций, I_k - накапливаемая информационная модель решаемой задачи.
- 3) Производится вычисление значения $f_A(\lambda_k)$.
- 4) Происходит пересчет информационной модели с учетом шага 3.

- 5) Анализ I_k и формирование конфигурации λ_{k+1} .
- 6) Проверка критерия остановки. Если критерий остановки выполняется - генерируем ответ λ^* , иначе полагаем $k := k + 1$ и переходим на шаг 3.