«Московский физико-технический институт (национальный исследовательский университет)
Физтех-школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»

Филатов Андрей Викторович

Быстрая оптимизация мультизадачных моделей

03.03.01 — Прикладные математика и физика

Выпускная квалификационная работа бакалавра

Научный руководитель:
д. ф.-м. н. Стрижов Вадим Викторович

Москва
2021

Аннотация

Мультизадачное обучение — область машинного обучения, рассматривающая одновременное решение нескольких задач. Особенностью мультизадачного обучения является необходимость учета взаимодействия между задачами. Создание модели мультизадачного обучения является более сложной задачей чем создание модели, решающей одну задачу, так как решение одной задачи часто препятствует решению другой задачи. Исходя из теоретических результатов, сходимость градиентного спуска к оптимальной точке гарантирована, если размер шага выбирается с помощью линейного поиска, чтобы удовлетворить правилу Армихо. Линейный поиск шага неэффективен из-за больших вычислительных затрат. В данной работе предлагается новая метод для алгоритмов линейного поиска в мультизадачном обучении, которая использует структурные свойства мультизадачных моделей. Идея была проверена на fast backtracking line search (FBLS). Проведено сравнение предложенного алгоритма с классическим backtracking line search (BLS) и градиентными методами с постоянной шагом на задачах MNIST, CIFAR-10, Cityscapes. Систематическое эмпирическое исследование показало, что предложенный метод приводит к большей производительности чем классический линейный поиск, а также сохраняет конкурентоспособное время и производительность по сравнению с градиентным спуском с постоянной шагом.

Ключевые слова: мультизадачное обучение, правило Армихо, линейный поиск

Содержание

1 Введение

Мультизадачное обучение [1] — один из подходов transfer learning [2], в котором строятся модели, которые умеют решать несколько задач одновременно. Достоинством этого подхода является улучшение качества модели на всех задачах за счет использования информации, содержащейся в обучающих сигналах связанных задач. Оно достигается за счет параллельного обучения задач с использованием общего представления, в результате чего задачи могут обмениваться информацией, что приводит к улучшению процесса обучения. Еще одним достоинством мультизадачного обучения является возможность уменьшать число параметров и повышать скорость работы модели за счет использование общих модулей. Многие подходы мультизадачного обучения показали свою эффективность и высокое качество во многих областях, таких как авиаконструирование, планирование радиотерапевтического лечения [3], компьютерное зрение [4], обработка естественного языка [5], обработка аудиосигналов [6].

Нейронные сети — это современный метод для решения различных задач машинного обучения, и мультизадачное обучение не является исключением. Оптимизация нейронных сетей часто использует методы градиентного спуска. Для градиентного спуска в случае мультизадачных моделей используется несколько подходов. Первый подход — скаляризация [7]. В этом случае решение нескольких задач сводится к решению одной задачи. Классическим методом скаляризации является взвешивание — рассмотрение мультизадачной проблем как взвешенной комбинации однозадачных проблем $\mathcal{L}^t$:

$$\mathcal{L} = \sum_{t=1}^{T} w_t \mathcal{L}^t; \quad w_t > 0.$$

Проблема взвешивания в том, что этот метод позволяет получить не все Парето оптимальные точки, что ограничивает возможности этого метода. Также этот подход сильно зависит от правильного выбора весов задач. В [8, 9] методы адаптивного взвешивания были введены для решения проблемы взвешивания. Второй подход - это метод min-max [10]. В этом случае мы ищем

направление $\boldsymbol{u}$, минимизирующее все функции равномерно:

$$\min_{d} \max_{t} (\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \boldsymbol{d})_t + g(\boldsymbol{d}).$$

Функция $g(\boldsymbol{d})$ имеет роль регуляризатора и обеспечивает единственность решения. Переход к двойственной задаче в методе min-max с $g(\boldsymbol{d}) = \|\boldsymbol{d}\|^2$ позволил построить метод MGDA[11].

Классические методы адаптивного шага, такие как Adam, NAG, RMSProp, AdaGrad, Adadelta могут быть применены для мультизадачного обучения и хорошо работают на практике, но не имеют теоретических гарантий сходимости в случае многокритериальной оптимизации. Использование методов линейного поиска имеет теоретические гарантии сходимости, но требует дополнительных вызовов функций, что в случае нейронных сетей имеет большую стоимость. Поэтому backtracking line search (BLS) [12, 13, 14, 15], метод золотого сечения и параболическая интерполяции неэффективны для нейронных сетей. Особенно это непрактично для мультизадачного обучения, когда есть необходимо делать проходы через несколько декодировщиков.

На основе структурных свойств многозадачных моделей с hard parameter sharing, предлагается новый подход для линейного поиска в случае мультизадачных моделей. Основная идея заключается в том, чтобы во время подбора шага выполнять проходы только по декодировщикам, что значительно сокращает затраты времени на поиск шага градиентного спуска поскольку кодировщик, содержащий большую часть вычислительные затрат, не используется для поиска шага. Предложенный подход быстрого линейного поиска (fast line search или FLS) был протестирован на задачах Multi-MNIST [16], CIFAR-10 и Cityscapes [17] с fast backtracking line search (FBLS), классическим backtracking line search (BLS) и градиентным спуском. По сравнению с BLS были получены более быстрая сходимость и более точное решение. По сравнению с градиентным спуском с фиксированным шагом был получен такой же уровень качества модели без долгого поиска подходящего шага градиентного спуска. При этом времени на обучение увеличилось только на 20%.

Данная работа имеет следующую структуру. В разделе 2 будет проведен обзор литературы. В разделе 3 будут введены основные понятия необходимые для понимания теории мультизадачных обучения. В разделе 4 будет описан подход быстрого линейного поиска и доказана теорема о сходимости данного подхода. В разделе 5 будут представлены результаты экспериментов.

2 Обзор литературы

Для полноценного обзора современного многозадачного обучения можно обратиться к [18, 19]. Классическим подходом к решению многокритериальной оптимизации является скаляризация [7]. Скаляризация — это сведение задачи многокритериальной оптимизации к задаче однокритериальной оптимизации. Существует несколько методов, позволяющих это сделать. Первый — взвешивание [20]. В этом случае минимизируется взвешенная комбинация задач. Другой подход — минимизация на симплексе, построенном на минимумах каждой задачи. Эти подходы включают метод нормального пересечения границ [21], метод нормальных ограничений [22]. Существует подходы с использованием эволюционных алгоритмов [23, 24]. Также применяются методы байесовской оптимизации [25, 26]. В них функции заменяются аппроксимациями и на них уже происходит многокритериальная оптимизация либо максимизируется объем в пространстве задач, чтобы приблизить Парето фронт. В [27] было добавлено разделение на области, где ищутся Парето стационарные точки, чтобы получить различные решения.

Другой подход основан на технике min-max. В [10] вводится min-max подход для стандартного градиентного спуска с L^2 регуляризацией, в результате чего находится вектор, минимизирующий все задачи равномерно. Далее метод min-max был обобщен на метод Ньютона [28] за счет выбора регуляризатора $g(\boldsymbol{d}) = \boldsymbol{d}^T \boldsymbol{H}^{\boldsymbol{t}} \boldsymbol{d}$, на стохастический градиентный спуск [29] и были получены скорости сходимости градиентных методов [30].

Первоначально двойственная задача к вышеупомянутому min-max подходу задача минимизации нормы выпуклой оболочки градиентов [11] стала новой техникой в многокритериальной оптимизации. В дальнейших работах метод был модифицирован путем добавления процесса Грама-Шмидта [31], модификации процесса Грама-Шмидта для получения быстрого решения [32] и модификации для методов второго порядка через нормализацию [32]. Также была доказана сходимость в стохастическом случае [33]. В [34] метод MGDA был применен к скрытому пространству, а не к пространству параметров. Предложенный подход позволяет оптимизировать верхнюю границу

классического MGDA (MGDA UpperBound). Добавление нормализации градиентов [35] в минимизацию нормы создает более устойчивое решение в случае несбалансированных задач. В [36] была добавлена регуляризация, чтобы устранить конфликты между задачами посредством ортогонализации градиентов различных задач. Последний метод, который можно классифицировать как метод выбора градиента — PCGrad [37]. В этом методе начальный конус решения изменяется на правильный «двойственный» конус, где каждый вектор из этого конуса минимизирует все функции потерь, и выбирается средний вектор этого конуса.

Также были предложены методы адаптивного взвешивания для преодоления проблемы поиска правильных весов. В [9] была использована дисперсия из оценки максимального правдоподобия в качестве весов. В [8] веса учитывали значения функций потерь для равномерной минимизации. Также в [38] было использовано взвешивание на основе функции потерь на предыдущем и текущем шаге.

Отдельным направлением в многозадачном обучении является выбор архитектуры. Выбор подходящей архитектуры позволяет внести дополнительную информацию и знания, которые улучшат процесс обучения и качество модели. В [39, 40] предположили, что общие параметры имеют тензорную структуру и рассмотрели применение методов тензорных факторизаций. В [41, 42, 43] были использованы блоки маршутизаторы. Эти блоки позволяют контролировать обмен информацией между промежуточными выходами кодировщика, чтобы стимулировать положительный трансфер знаний и предотвращать негативный. В [44, 45] для построения многозадачной модели был рассмотрены подходы поиска нейросетевых архитектур. В [46, 47, 38, 48] были использованы различные механизмы внимания и адаптеры. Адаптеры имеют меньше параметров чем основная модель, и позволяют решать не только многозадачных проблемы, но и мультидоменные проблемы. В [49] было проведено исследование взаимосвязей между различными задачами компьютерного зрения. В результате был построена таксономия, показывающая какие задачи стоит решать вместе, а какие нет.

3 Многокритериальная оптимизация

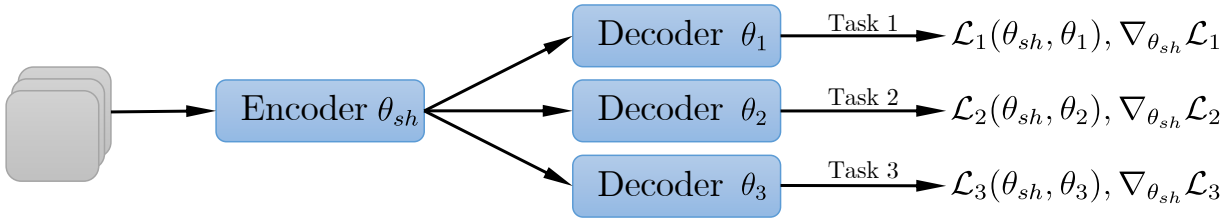

Рис. 1: Пример мультизадачной модели с hard parameter sharing

В работе будут использоваться следующие обозначения:

- X — признаковое пространство и $Y = Y^1 \times \ldots \times Y^T$ - пространство задач.
- $(\boldsymbol{\theta}^{sh})$ — разделяемые параметры и $(\boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T)$ — параметры специфичные для задачи.
- $\mathbf{z}(\boldsymbol{x}, \boldsymbol{\theta}^{sh})$ — модель кодировщика, а Z — скрытое пространство, созданное кодировщиком.
- Выборка $\{\boldsymbol{x}_i, \boldsymbol{y}_i^1, \ldots, \boldsymbol{y}_i^T\}_{i=1}^n$.
- Декодировщик для задачи t: $f^t(\boldsymbol{z}(\boldsymbol{x}, \boldsymbol{\theta}^{sh}), \boldsymbol{\theta}^t) = f^t(\boldsymbol{z}, \boldsymbol{\theta}^t) : Z \to Y^t$.
- $\boldsymbol{d}_z$ и $\boldsymbol{d}_{sh}$ — некоторые векторы в скрытом пространстве и пространстве параметров соответственно, которые удовлетворяют следующим условиям:

$$\forall t = \overline{1, T} \quad \nabla_z^T \mathcal{L}^t \boldsymbol{d}_z < 0$$

$$\forall t = \overline{1, T} \quad \nabla_{\boldsymbol{\theta}^{sh}}^T \mathcal{L}^t \boldsymbol{d}_{sh} < 0$$

Это условие означает, что применение шага градиентного спуска по направлению $\boldsymbol{d}_z$ в скрытом пространстве или по направлению $\boldsymbol{d}_{sh}$ в пространстве параметров минимизирует все функции.

- Функция потерь для задачи t:

$$\hat{\mathcal{L}}^t\left(\boldsymbol{z}, \boldsymbol{\theta}^t\right) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}^t\left(f^t\left(\mathbf{z}(\boldsymbol{x}_i, \boldsymbol{\theta}^{sh}), \boldsymbol{\theta}^t\right), \boldsymbol{y}_i^t\right)$$

Рассмотрим задачу многозадачного обучения с T задачами:

$$\min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \mathbf{L}\left(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T\right) = \min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \left(\hat{\mathcal{L}}^1\left(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1\right), \ldots, \hat{\mathcal{L}}^T\left(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^T\right)\right)^\top \tag{3.1}$$

Для решения этой задачи многозадачного обучения рассмотрим частичный порядок — Парето доминирование .

Определение 3.1. *Точка $\boldsymbol{\theta}_1$ Парето доминирует точку $\boldsymbol{\theta}_2$, если:*

$$\forall t \in \{1, \ldots, T\}\ \hat{\mathcal{L}}^t(\boldsymbol{\theta}_2) \leqslant \hat{\mathcal{L}}^t(\boldsymbol{\theta}_1)$$

$$\exists i : \hat{\mathcal{L}}^i(\boldsymbol{\theta}_2) < \hat{\mathcal{L}}^i(\boldsymbol{\theta}_1)$$

Используя частичный порядок, можно свести решение задачи мультизадачного обучения к поиску Парето оптимальной точки.

Определение 3.2. *Точка $\hat{\boldsymbol{\theta}}$ в задаче 3.1 является Парето оптимальной (или Парето эффективной) тогда и только тогда, когда она не доминируется никакой другой точкой.*

В частности, если точка $\boldsymbol{\theta}$ не является Парето оптимальной, то существует по крайней мере одна точка, которая Парето доминирует ее [50].

Для Парето оптимальных точек нет удобного критерия, по которому их можно искать. Но, в случае когда функции $\hat{\mathcal{L}}^t$ — непрерывно-дифференцируемы, то можно записать необходимое условие Парето оптимальности — Парето стационарность [34, 11].

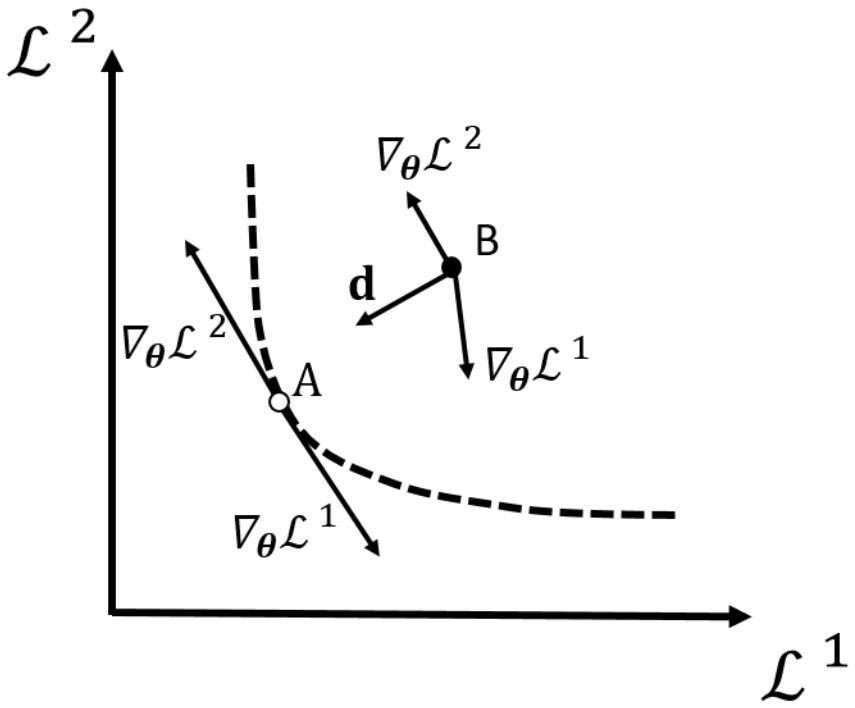

Рис. 2: Визуализация Парето стационарности. Точка A является Парето стационарной. Точка B не является Парето стационарной, поэтому для нее существует минимизирующее направление $\boldsymbol{d}$.

Определение 3.3. *Точка $\hat{\boldsymbol{\theta}}$ является Парето стационарной точкой для задачи 3.1, если существуют веса $\alpha^1, \ldots, \alpha^T \geqslant 0$ такие, что*

- $\sum_{t=1}^{T} \alpha^t = 1.$
- $\sum_{t=1}^{T} \alpha^t \nabla_{\boldsymbol{\theta}^{*h}} \hat{\mathcal{L}}^t \left(\hat{\boldsymbol{\theta}}^{sh}, \hat{\boldsymbol{\theta}}^t \right) = 0.$
- $\forall t \ \nabla_{\boldsymbol{\theta}^t} \hat{\mathcal{L}}^t \left(\hat{\boldsymbol{\theta}}^{sh}, \hat{\boldsymbol{\theta}}^t \right) = 0.$

4 Быстрый линейный поиск

В этом разделе описывается идея использования скрытого пространства вместо пространства общих параметров для линейного поиска шага градиентного спуска. Оптимизация в скрытом пространстве более вычислительно эффективна, так как скрытое пространство имеет меньшую размерность. Также оптимизация в скрытом пространстве позволяет не использовать кодировщик для нахождения шага.

Рассмотрим градиенты: $\nabla_{\boldsymbol{\theta}^{sh}} L^t, \nabla_{\boldsymbol{z}} L^t$. Чтобы получить направление минимизации $\boldsymbol{d}_{sh}$ или $\boldsymbol{d}_z$ можно использовать методы PCGrad [37], или EDM [35], или MGDA [11, 34]. Суть алгоритма линейного поиска заключается в том, чтобы найти шаг η, чтобы:

$$\mathcal{L}^t(\boldsymbol{\theta} - \eta \boldsymbol{d}) < \mathcal{L}^t(\boldsymbol{\theta}),\ \forall t \in \{1 \dots T\}. \tag{$\star$}$$

Для линейного поиска существует следующая теорема.

Теорема 4.1. *Если условие ($\star$) будет выполнено на каждой итерации, то для любой сходящейся подпоследовательности $\{\boldsymbol{\theta}_{\boldsymbol{k_j}}\}_{j=1}^{\infty} : \lim_{j\to\infty} \boldsymbol{\theta}_{\boldsymbol{k_j}} = \hat{\boldsymbol{\theta}}$, созданной градиентным спуском, предел этой последовательности $\hat{\boldsymbol{\theta}}$ — Парето станционарная точка.*

Доказательство этой теоремы приведено в [10]. Чтобы обеспечить выполнения условия ($\star$), используется правило Армихо, где $\beta \in (0, 0.5)$:

$$\forall t \in \{1, \dots, T\} :\ \mathcal{L}^t(\boldsymbol{\theta}^{sh} - \eta \boldsymbol{d}_{sh}, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathcal{L}^t) \leqslant \mathcal{L}^t - \eta\beta \left\| \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 - \eta\beta \left(\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^{sh}} \right)^{\top} \boldsymbol{d}_{sh} \tag{4.1}$$

В классическом линейном поиске (Алгоритм 4.1) необходимо обновлять общие параметры несколько раз на каждой итерации. Поэтому перейдем к рассмотрению скрытого пространства посредством линейной аппроксимации:

$$\boldsymbol{z}(\boldsymbol{\theta}^{sh} - \eta \boldsymbol{d}_{sh}) \approx \boldsymbol{z} - \eta \boldsymbol{d}_z$$

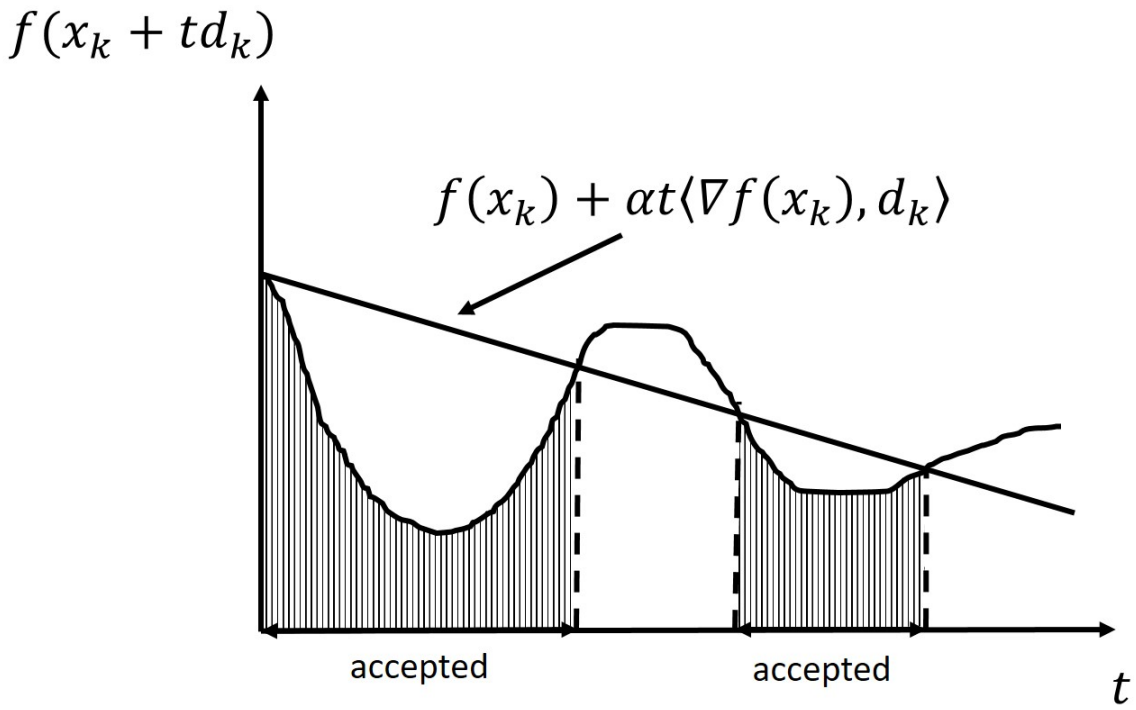

Рис. 3: Визуализация правило Армихо. Условие Армихо выполняется, когда функция находится ниже прямой

Алгоритм 4.1 Backtracking line search

Вход: β, γ, lr_{ub}
Выход: Learning rate η

повторять
  $\eta \leftarrow \gamma \cdot \eta$
  $\boxed{\tilde{\boldsymbol{\theta}}^{sh} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot \boldsymbol{d}_{sh}}$
  для $t \leftarrow 1$ to T
    $\tilde{\boldsymbol{\theta}}^t \leftarrow \boldsymbol{\theta}^t - \eta \cdot \nabla_{\boldsymbol{\theta}^t} L^t$
пока правило Армихо (4.1)
для $t \leftarrow 1$ to T
  $\boldsymbol{\theta}^t_{new} \leftarrow \tilde{\boldsymbol{\theta}}^t$
$\boxed{\boldsymbol{\theta}^{sh}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{sh}}$

Тогда можно рассмотреть модифицированное правило Армихо:

$$\forall t \in \{1, \ldots, T\}: \ \mathcal{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathcal{L}^t) \leqslant \mathcal{L}^t - \eta\beta \left\| \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 - \eta\beta \left(\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{z}} \right)^T \boldsymbol{d}_z \quad (4.2)$$

Алгоритм 4.2 Fast backtracking line search (Ours)

Вход: β, γ, lr_{ub}
Выход: Learning rate η

повторять
  $\eta \leftarrow \gamma \cdot \eta$
  $\boxed{\boldsymbol{z} \leftarrow \boldsymbol{z} - \eta \cdot \boldsymbol{d}_z}$
  для $t \leftarrow 1$ to T
    $\tilde{\boldsymbol{\theta}}^t \leftarrow \boldsymbol{\theta}^t - \eta \cdot \nabla_{\boldsymbol{\theta}^t} L^t$
пока правило Армихо (4.2)
для $t \leftarrow 1$ to T
  $\boldsymbol{\theta}^t_{new} \leftarrow \tilde{\boldsymbol{\theta}}^t$
$\boxed{\boldsymbol{\theta}^{sh}_{new} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot \frac{\partial \boldsymbol{\theta}^{sh}}{\partial \boldsymbol{z}} \boldsymbol{d}_z}$

С модифицированным правилом Армихо алгоритма линейного поиска будеть иметь вид (Алгоритм 4.2).

В следующей теореме показывается, что градиентному спуску с быстрым линейным поиском сходится к Парето стационарной точке.

Теорема 4.2. *Каждый частичный предел последовательности* $\boldsymbol{\theta}_k = [\boldsymbol{\theta}^{sh}_k, \boldsymbol{\theta}^1_k, \ldots, \boldsymbol{\theta}^t_k]_{k=1}^{\infty}$, *порожденной алгоритмом 4.2 со правилами Армихо* (4.3), (4.4), (4.5) *является Парето стационарной точкой:*

$$\mathcal{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathcal{L}^t) \leqslant \mathcal{L}^t - \eta\beta \left\| \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 \quad \forall t \in \{1 \ldots T\} \tag{4.3}$$

$$\mathcal{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathcal{L}^t) \leqslant \mathcal{L}^t - \eta\beta \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t}^\top \boldsymbol{d}_z \quad \forall t \in \{1 \ldots T\} \tag{4.4}$$

$$\mathcal{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathcal{L}^t) \leqslant \mathcal{L}^t - \eta\beta \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t}^\top \boldsymbol{d}_z - \eta\beta \left\| \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 \quad \forall t \in \{1 \ldots T\} \tag{4.5}$$

Доказательство.

1. Пусть $\boldsymbol{u}^t = \nabla_{\boldsymbol{\theta}^t} L^t$. Пусть $\bar{\boldsymbol{\theta}}$ — частичный предел последовательности $\boldsymbol{\theta}_k$. Тогда, существует подпоследовательность $\{\boldsymbol{\theta}_{k_j}\}_{j=1}^{\infty}$ сходящаяся к $\bar{\boldsymbol{\theta}}$. Так как $\boldsymbol{L}$ непрерывна: $\boldsymbol{L}(\boldsymbol{\theta}) \to \boldsymbol{L}(\bar{\boldsymbol{\theta}})$. Следовательно, $\eta\beta\|\boldsymbol{u}^t\|^2 \to 0$. и существует следующая альтернатива:

   - $\limsup \eta > 0$

- $\lim \eta = 0$

В первом случае $\|\boldsymbol{u}^t\|^2 \to 0$, следовательно, $\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{z}} = \boldsymbol{u}^t \frac{\partial \boldsymbol{\theta}^t}{\partial \boldsymbol{z}} \to 0$ и согласно [34] $\bar{\boldsymbol{\theta}}$ — Парето стационарная точка.

Во втором случае предположим, что частичный предел $\bar{\boldsymbol{\theta}}$ не является Парето стационарной точкой. Тогда, существует $\bar{\boldsymbol{d}}$ — минимизирующее направление. Так как $\lim \eta = 0$, то для любого константного шага η_n, начиная с некоторого j_0 условие Армихо не выполняется по крайней мере для одной функции:

$$\forall \eta_n = \frac{1}{n} \; \exists j_0 : \forall j \geqslant j_0 \; \exists t_n \; \mathcal{L}^{t_n}(\boldsymbol{z}_{k_j} - \eta_n \boldsymbol{d}_{k_j}, \boldsymbol{\theta}_{k_j} - \eta_n \boldsymbol{u}_j^{t_n}) \geqslant \mathcal{L}^{t_n}(\boldsymbol{z}_{k_j}) - \eta_n \beta \|\boldsymbol{u}_j^{t_n}\|^2$$

Так как последовательность индексов задач $\{t_n\}_{n=1}^{\infty}$ ограничена числом задач T, то можно выделить подпоследовательность $\{t_{n_m}\}_{m=1}^{\infty}$, которая сходится к некоторому индексу t_0.

Следовательно, начиная с некоторого j_0 для $\mathcal{L}^{t_0}$ верно следующее:

$$\forall j \geqslant j_0 \quad \mathcal{L}^{t_0}(\boldsymbol{z}_{k_j} - \eta_n \boldsymbol{d}_{k_j}, \boldsymbol{\theta}_{k_j}^{t_0} - \eta_n \boldsymbol{u}_j^{t_0}) \geqslant \mathcal{L}^{t_0}(\boldsymbol{z}_{k_j}) - \eta_n \beta \|\boldsymbol{u}_j^{t_0}\|^2$$

Так как $\boldsymbol{L}$ непрерывно-дифференцируема и $\boldsymbol{\theta}_{k_j} \to \bar{\boldsymbol{\theta}}$, то $\boldsymbol{d}_{k_j} \to \bar{\boldsymbol{d}}$ и можно получить:

$$\mathcal{L}^{t_0}(\bar{\boldsymbol{z}} - \eta_n \bar{\boldsymbol{d}}, \bar{\boldsymbol{\theta}}^{t_0} - \eta_n \bar{\boldsymbol{u}}^{t_0}) \geqslant \mathcal{L}^{t_0}(\bar{\boldsymbol{z}}) - \eta_n \beta \|\bar{\boldsymbol{u}}^{t_0}\|^2$$

Так как это верно $\forall \eta_n = \frac{1}{n}$, где $n \in \mathbb{N}$, то мы получаем противоречие с правилом Армихо: так как $\bar{\boldsymbol{d}}$ — минимизирующее направление, то:

$$\exists \bar{\eta} : \; \forall t \in \{1, \ldots, T\} : \; L^t(\bar{\boldsymbol{z}} - \eta \bar{\boldsymbol{d}}, \bar{\boldsymbol{\theta}}^t - \eta \bar{\boldsymbol{u}}^t) \leqslant \mathcal{L}^t(\bar{\boldsymbol{z}}) - \eta \beta \|\boldsymbol{u}^t\|^2 - \eta \beta \frac{\partial \mathcal{L}^t}{\partial \boldsymbol{z}}^T s.$$

Поэтому $\bar{\boldsymbol{d}} = 0$ и $\bar{\boldsymbol{\theta}}$— Парето стационарная точка.

2. Для этого правила Армихо доказательство может быть получено при помощи следующей модификации. Так как $\boldsymbol{L}(\boldsymbol{\theta}_{k_j}) \to \boldsymbol{L}(\bar{\boldsymbol{\theta}})$, то

$\eta\beta\frac{\partial\mathcal{L}^t}{\partial \boldsymbol{z}}^T\bar{\boldsymbol{d}} \;\to\; 0.$

В первом случае альтернативы $\forall t \quad \frac{\partial\mathcal{L}^t}{\partial \boldsymbol{z}}^T\bar{\boldsymbol{d}} = 0$. Так как матрица $\|\frac{\partial\mathcal{L}^t}{\partial z}\|_{t=1}^T$ невырожденная, то $\bar{\boldsymbol{d}} = 0$ и $\bar{\boldsymbol{\theta}}$ — Парето стационарная точка по [34].

Во втором случае альтернативы можно заменить $\eta\beta\|\boldsymbol{u}_{t_0}\|^2$ на $\eta\beta\frac{\partial\mathcal{L}^t}{\partial \boldsymbol{z}}^T\boldsymbol{d}$ и доказательство не изменится.

3. Для этого правила Армихо доказательство может быть получено следующим образом. Так как $\boldsymbol{L}(\boldsymbol{\theta}_{k_j}) \to \boldsymbol{L}(\bar{\boldsymbol{\theta}})$, то $\eta\beta\frac{\partial\mathcal{L}^t}{\partial \boldsymbol{z}}^T\bar{\boldsymbol{d}} + \eta\beta\|\boldsymbol{u}^t\|^2 \to 0$. Оба члена неотрицательны поэтому имеется исходная альтернатива.

   В первом случае у нас имеется, что $\bar{\boldsymbol{d}} = 0, \boldsymbol{u}^t = 0$, то $\bar{\boldsymbol{\theta}}$ — Парето стационарная точка по [34].

   Во втором случае мы можем заменить $\eta\beta\|\boldsymbol{u}^t\|^2$ на $\eta\beta\frac{\partial L^t}{\partial \boldsymbol{z}}^T\bar{s} + \eta\beta\|\boldsymbol{u}^t\|^2$ и доказательство не изменится.

■

5 Эксперименты

Таблица 1: Сравнение времен работы алгоритмов. В скобках указано число секунд на одну эпоху.

	MNIST	CIFAR-10	Cityscapes
Fast backtracking (Ours)	1.05 (143)	0.15 (85)	1.28 (76800)
Backtracking	1.37 (195)	1.18 (650)	-
Classical SGD	1.0 (143)	1.0 (550)	1.0 (60000)
MGDA-UB [34]	0.95 (136)	0.14 (80)	-

Для сравнения были рассмотрены быстрый бэктрекинг, классический бэктрекинг, классический SGD и MGDA-UB [34]. Эксперименты проводились на задачах MultiMNIST, CIFAR-10 и Cityscapes. Для backtracking нижнюю граница шага градиентного спуска была выбрана как $\varepsilon = 10^{-10}$ для вычислительной устойчивости, а параметр $\beta = 0.1$. Начальная верхняя граница была установлена как $\eta = 1$. Из-за стохастичности градиентов верхняя граница размера шага уменьшалась на $\gamma = 0.5$ каждые $N = 10$ эпох.

5.1 MultiMNIST

Датасет MultiMNIST был сгенерирован из датасета MNIST следующим образом. Были взяты два изображения из MNIST. Одно изображение было смещено на четыре пикселя влево, другое было смещено на четыре пикселя вправо и затем наложены друг на друга. Задача классифицировать изображение слева и изображение справа. Для обучения использовались 60000 изображений, для тестирования использовались 10000 изображений.

Для MultiMNIST была использована архитектура LeNet-5. Размер батча 256. Обучение проводилось 100 эпох. Для сравнения со стандартным градиентным спуском было выбрано восемь фиксированных шагов, равномерно распределенных в логарифмическом диапазоне между -3 и -1. В качестве функции потерь выбрана кросс-энтропия, а в качестве ошибки — (1 - точность). Результаты были усреднены по 5 экспериментам (Рисунок 5).

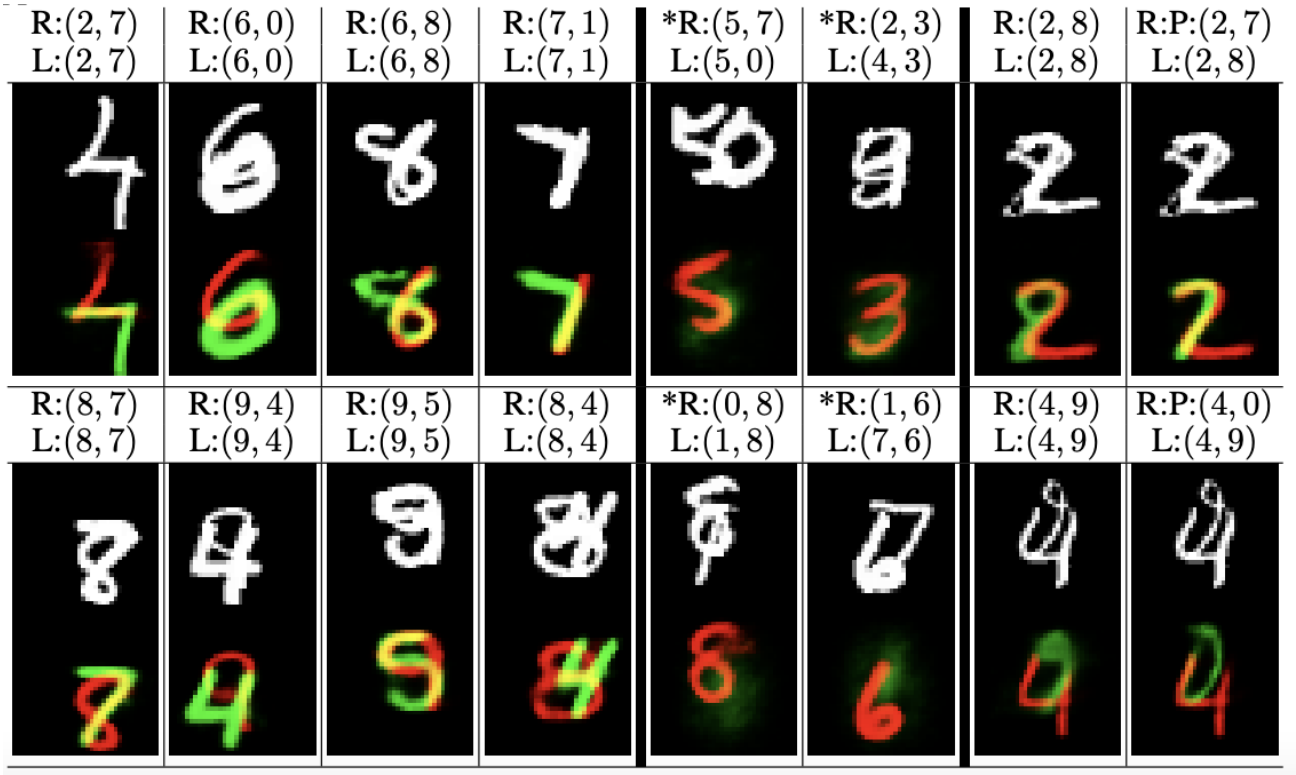

Рис. 4: Пример изображений MultiMNIST

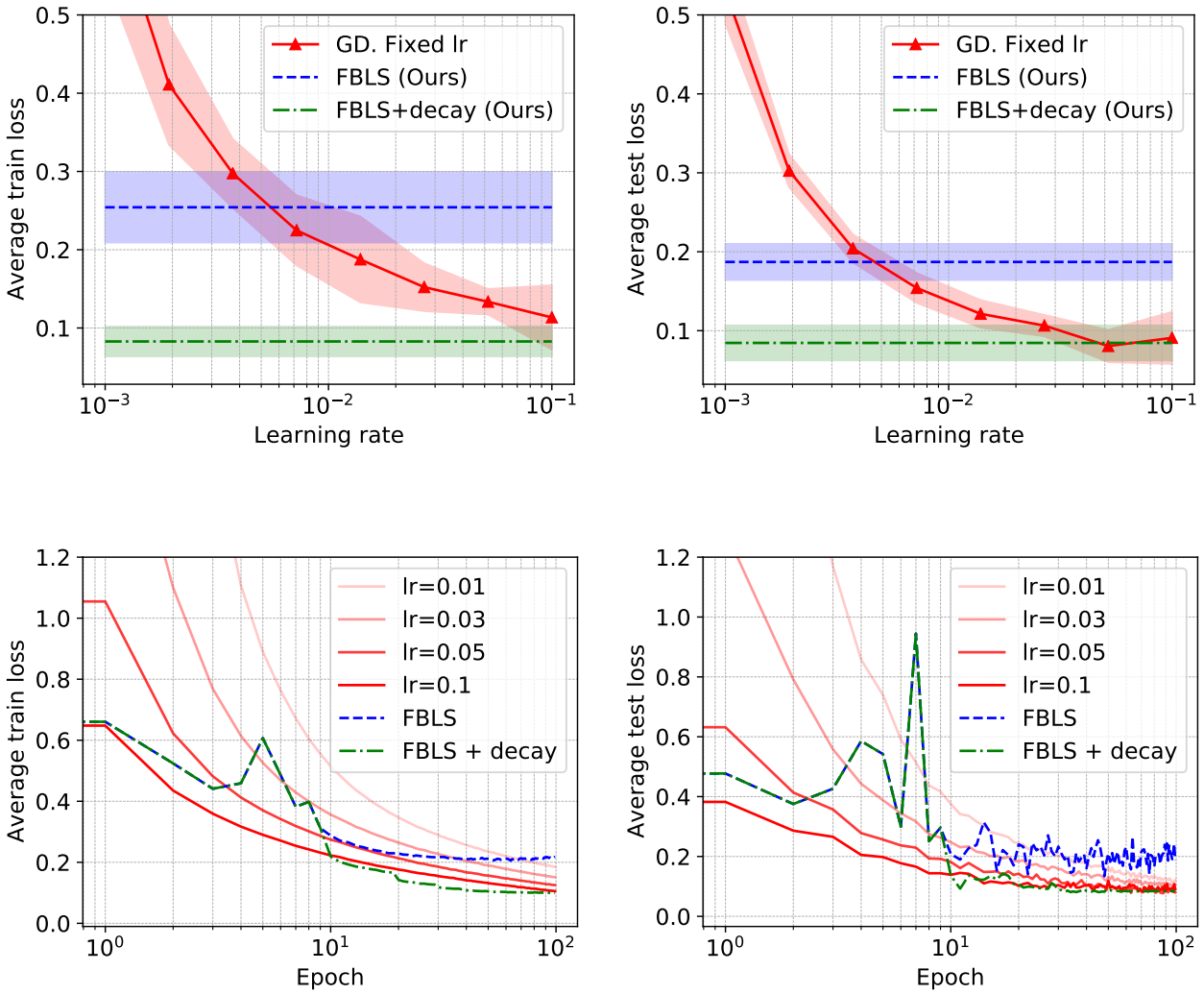

Рис. 5: Сравнение предложенного метода с методом градиентного спуска с фиксированной шагом градиентного спуска на MultiMNIST.

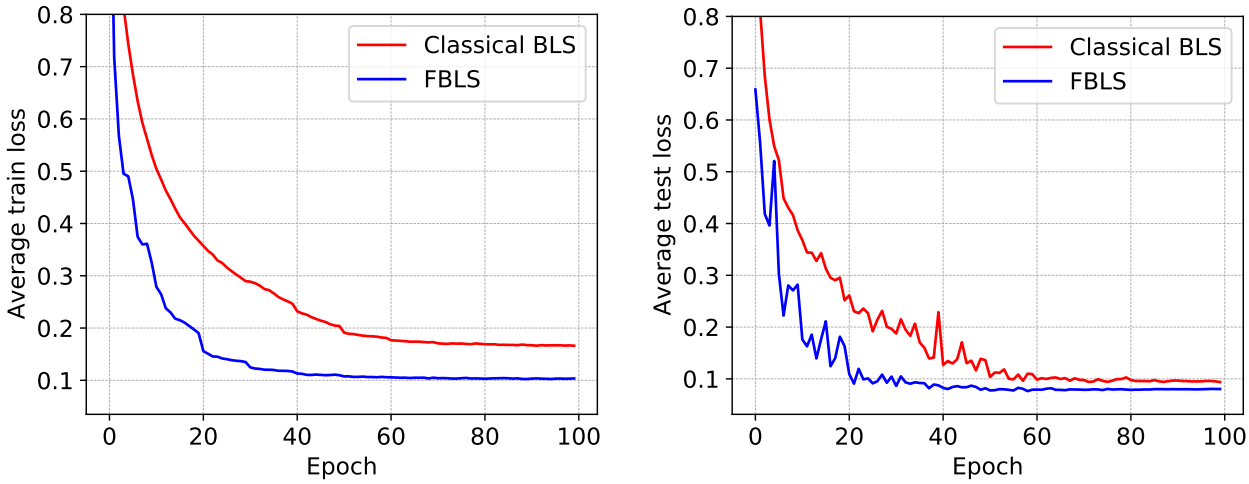

Рис. 6: Сравнение предложенного метода с классическим бэктрекингом Армихо на MultiMNIST.

5.1.1 Классический бэктрекинг

Для сравнения классического и быстрого линейного поиска с backtracking было рассмотрено их поведение при различных $\beta = \{0.1, 0.2, 0.3, 0.4\}$. Усредненные по β результаты представлены на рисунке 6.

По результатам экспериментов можно отметить, что быстрый бэктрекинг сходится быстрее и имеет лучшее качество. Также, при сравнении времени работ (Таблица 1) было выявлено, что быстрый бэктрекинг быстрее на 50% чем классический бэктрекинг.

5.2 CIFAR-10

Набор данных CIFAR-10 содержит 60000 цветных изображений 32×32 в 10 различных классах. 10 различных классов представляют собой самолеты, автомобили, птицы, кошки, олени, собаки, лягушки, лошади, корабли и грузовики. В каждом классе имеется 6000 изображений.

Для CIFAR-10 10 классов были использованы для создания 10 синтетических задач классификации "one vs rest". Размер батча был выбран равным 256. В качестве кодировщика была выбрана модель ResNet-18, а в качестве декодировщика один полносвязный слой. Результаты представлены на рисунке 8.

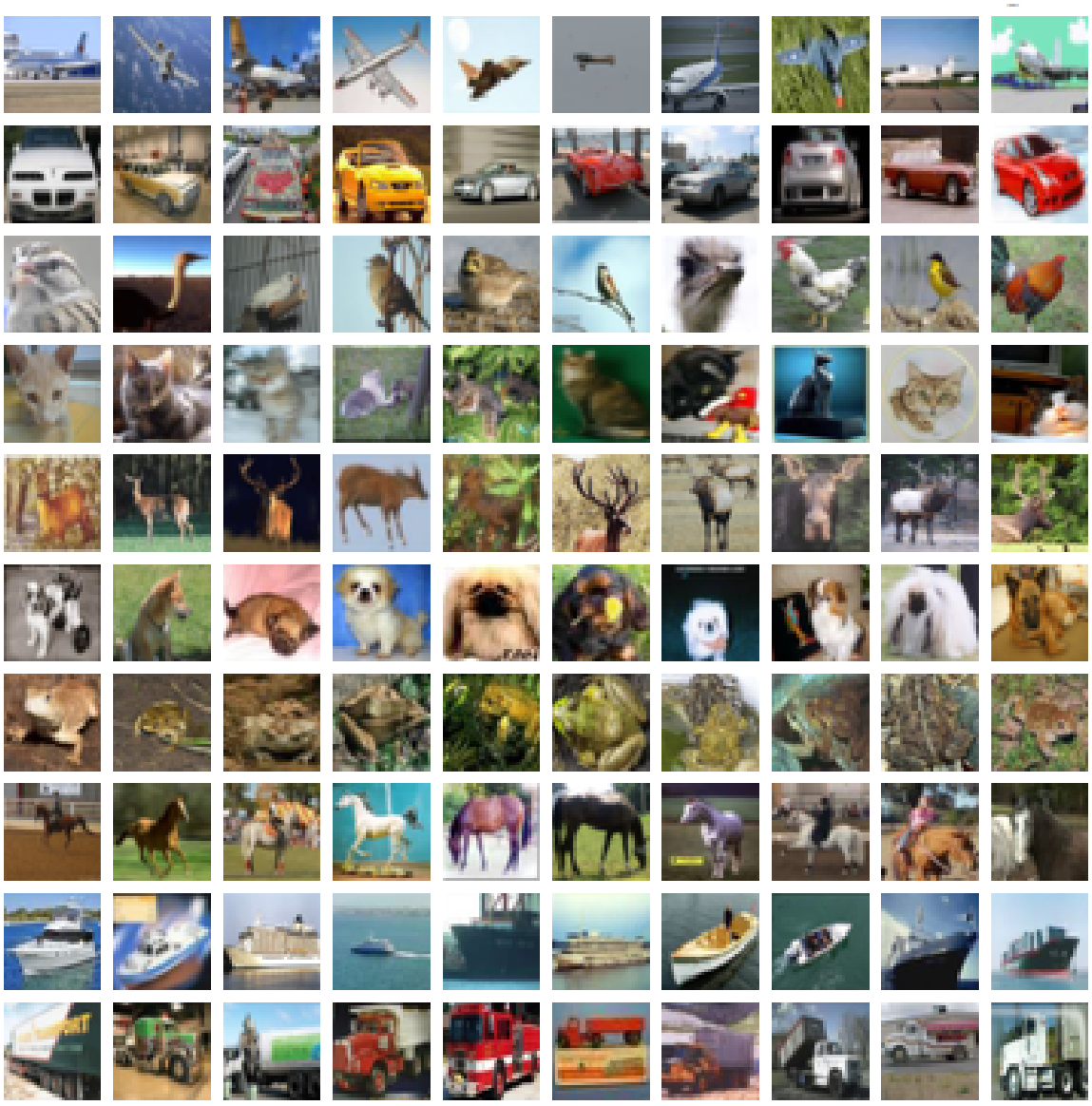

Рис. 7: Пример изображений CIFAR-10

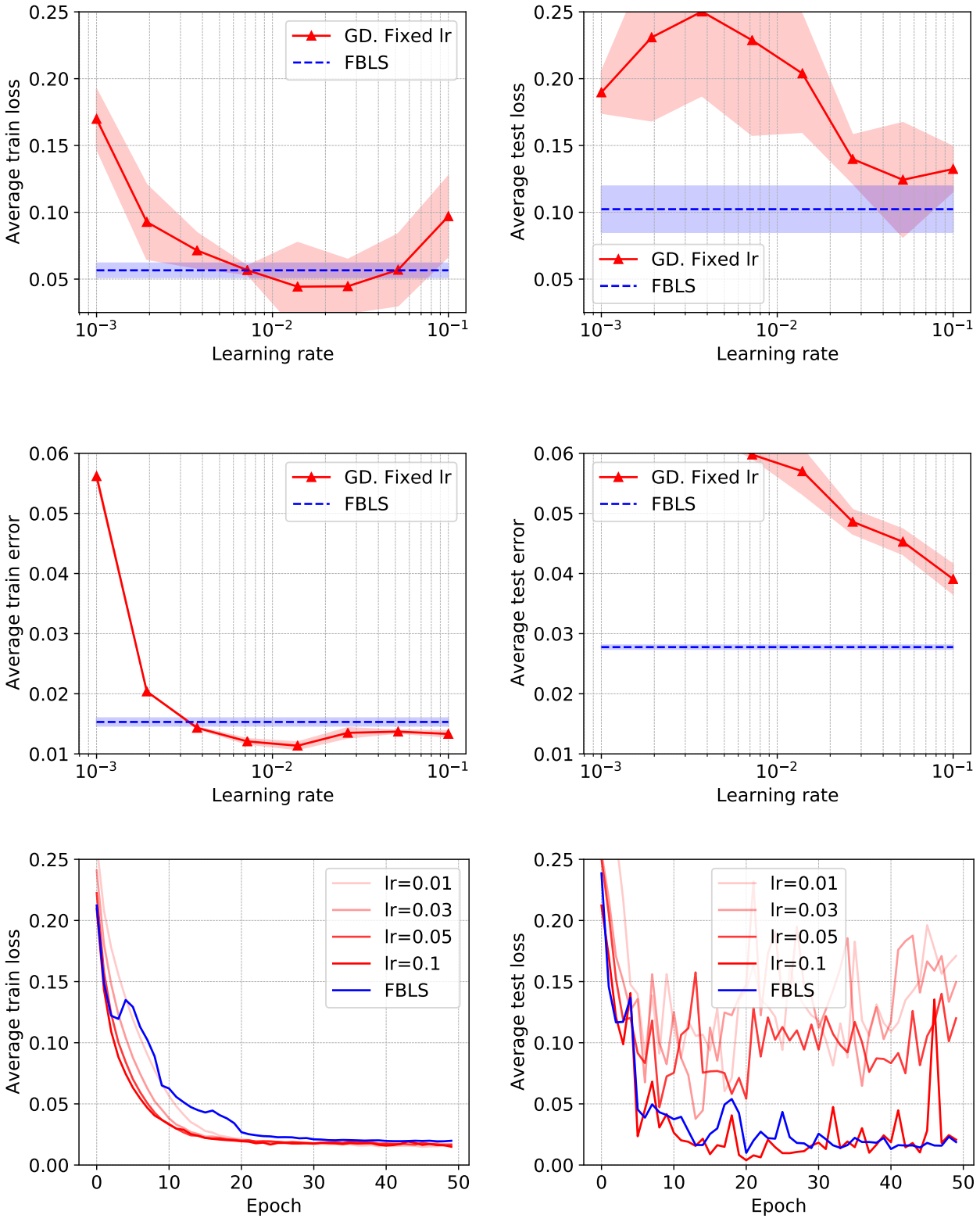

Рис. 8: Сравнение предложенного метода с методом градиентного спуска с фиксированной шагом градиентного спуска на CIFAR-10.

Рис. 9: Пример изображений Cityscapes

Для градиентного спуска восемь различных шагов были выбраны равномерно в логарифмическом диапазоне от -3 до -1. В качестве функции потерь на всех задачах использовалась кросс энтропия, а в качестве функции ошибки — (1 - точность).

5.3 Cityscapes

Cityscapes — это крупномасштабная база данных, ориентированная на семантическое понимание городских уличных сцен. Она представляет собой разметку пикселей для 30 классов, сгруппированных в 8 категорий (плоские поверхности, люди, транспортные средства, сооружения, объекты, природа, небо и пустота). Набор данных состоит из около 5000 точно размеченных изображений и 20000 грубо размеченных. Для Cityscapes решались три задачи: semantic segmentation, instance segmentation, disparity estimation.

В качестве кодировщика была выбрана модель ResNet-50, предобученная на Imagenet, а в качестве декодировщика PSPNet. Были сравнены градиентный спуск с постоянным шагом и быстрый бэктрекинг. Для градиентного спуска были рассмотрены следующие шаги $\{0.001, 0.01, 0.1\}$. Результаты представлены на рисунках 11, 12, 13.

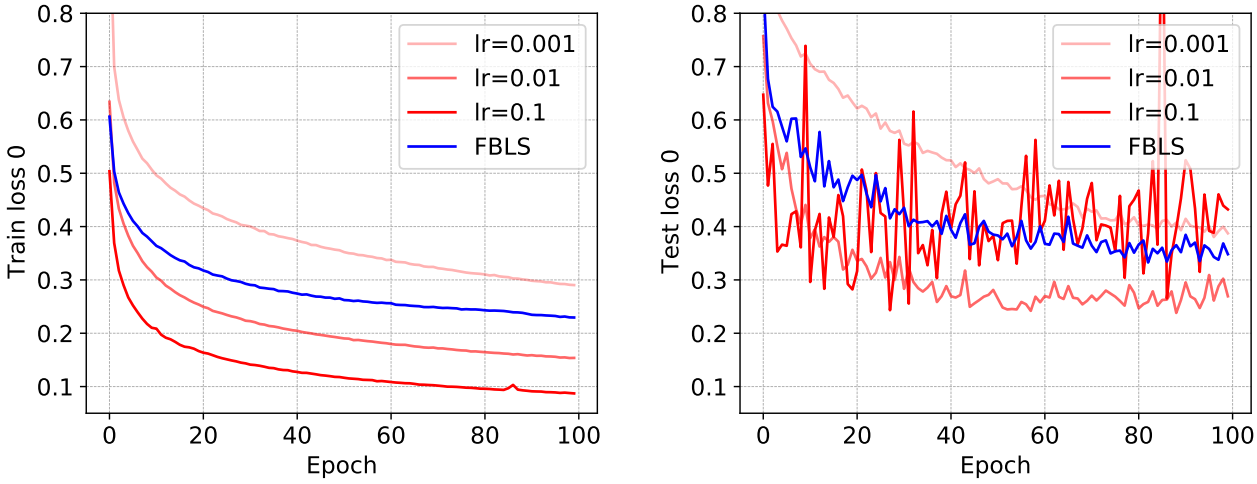

Рис. 10: Сравнение с градиентным спуском для semantic segmentation

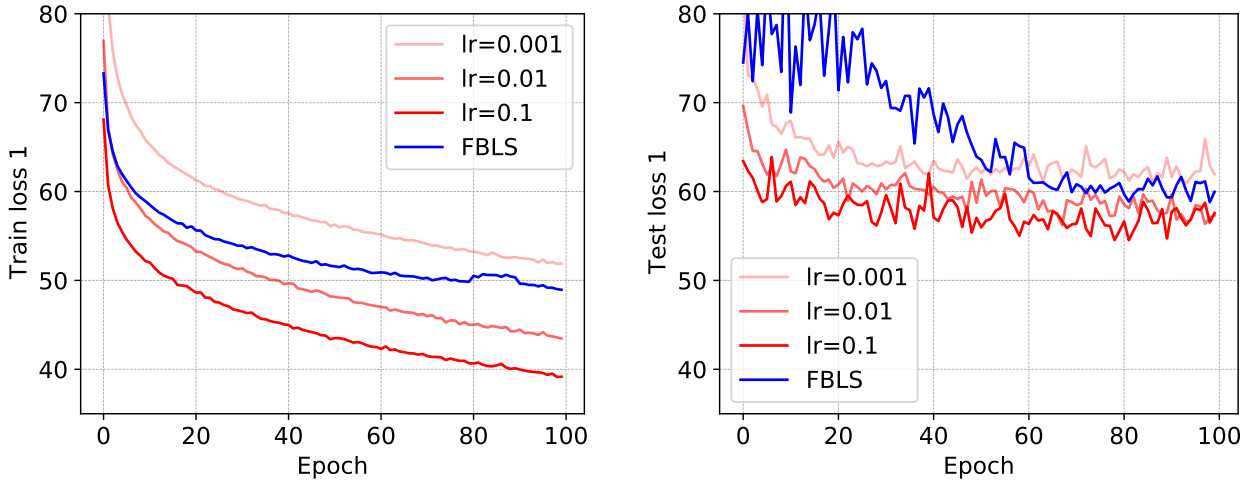

Рис. 11: Сравнение с градиентным спуском на задаче instance segmentation

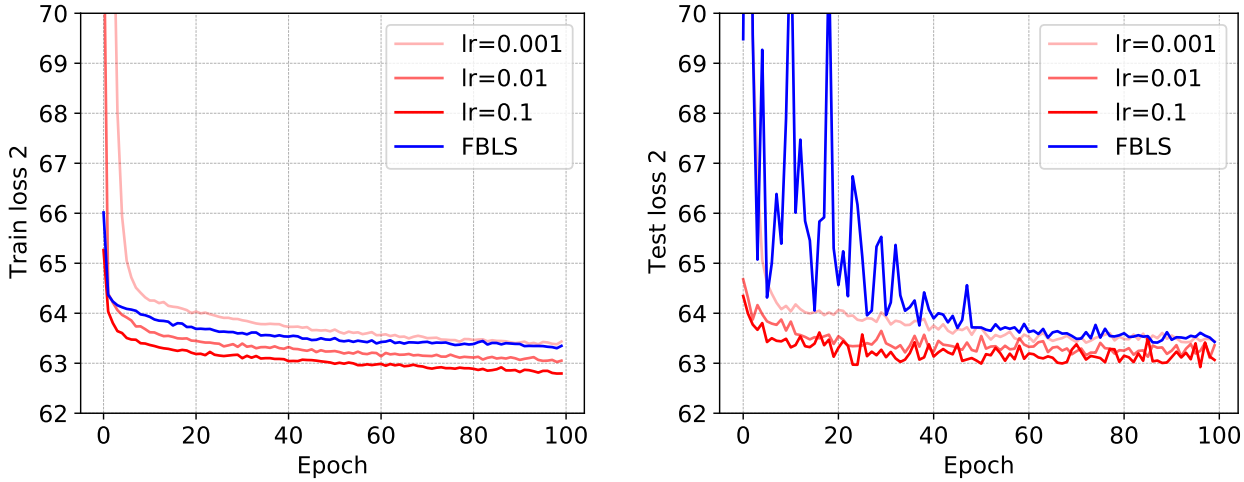

Рис. 12: Сравнение с градиентным спуском на задаче depth disparity

6 Заключение

Основные результаты работы

- Предложен метод оптимизации мультизадачных моделей;
- Подтверждена теоретическая сходимость предложенного метода;
- Проведены вычислительные эксперименты, которые подтвердили эффективность метода на задачах MultiMNIST, CIFAR-10, Cityscapes.

В будущих работах будет рассмотрены следующие вопросы.

Первый вопрос — сочетание линейного поиска с адаптивными методами градиентного спуска (Adam, Adagrad, RMSProp). Для этих методов будет проведен теоретический и экспериментальный анализ, чтобы выяснить применимость предложенного подхода.

Второй вопрос — уменьшение верхней границы шага обучения. Без этого метод становится неэффективным на практике. В качестве решения было предложено ручное уменьшение границы, но данный подход требует тщательного подбора гиперпараметров, связанных с параметрами линейного поиска. Поэтому в следующих работах будет исследовано возможность обнаружения ситуации, когда нужно уменьшить границу.

Третий вопрос — исследование градиентных методов более высокого порядка. В данной работе были рассмотрены только методы первого порядка. Однако, в общей теории мультизадачных моделей уже получены и доказаны результаты для методов второго порядка. Поэтому будет исследован вопрос обобщения предложенного подхода на случай методов второго порядка.

Список литературы

[1] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[2] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[3] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. Di Stefano, *Real-World Applications of Multiobjective Optimization*, pp. 285–327. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[4] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6129–6138, 2017.

[5] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," *arXiv preprint arXiv:1804.00079*, 2018.

[6] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, "Rapid adaptation for deep neural networks through multi-task learning," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[7] J. Johannes, "Scalarization in vector optimization," *Mathematical Programming*, vol. 29, no. 2, pp. 203–218, 1984.

[8] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *International Conference on Machine Learning*, pp. 794–803, PMLR, 2018.

[9] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

[10] J. Fliege and B. F. Svaiter, "Steepest descent methods for multicriteria optimization," *Mathematical Methods of Operations Research*, vol. 51, no. 3, pp. 479–494, 2000.

[11] J.-A. Désidéri, "Mgda variants for multi-objective optimization," 2012.

[12] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[13] P. Wolfe, "Convergence conditions for ascent methods," *SIAM review*, vol. 11, no. 2, pp. 226–235, 1969.

[14] S. Vaswani, A. Mishkin, I. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien, "Painless stochastic gradient: Interpolation, line-search, and convergence rates," in *Advances in Neural Information Processing Systems*, pp. 3732–3745, 2019.

[15] S. Vaswani, F. Kunstner, I. Laradji, S. Y. Meng, M. Schmidt, and S. Lacoste-Julien, "Adaptive gradient methods converge faster with over-parameterization (and you can do a line-search)," *arXiv preprint arXiv:2006.06835*, 2020.

[16] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, pp. 3856–3866, 2017.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[18] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[19] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[20] I. Das and J. E. Dennis, "A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems," *Structural optimization*, vol. 14, no. 1, pp. 63–69, 1997.

[21] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM journal on optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[22] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the pareto frontier," *Structural and multidisciplinary optimization*, vol. 25, no. 2, pp. 86–98, 2003.

[23] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.

[24] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii," in *International conference on parallel problem solving from nature*, pp. 849–858, Springer, 2000.

[25] S. Daulton, M. Balandat, and E. Bakshy, "Differentiable expected hypervolume improvement for parallel multi-objective bayesian optimization," 2020.

[26] T. Wada and H. Hino, "Bayesian optimization for multi-objective optimization and multi-point search," 2019.

[27] X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong, "Pareto multi-task learning," in *Advances in Neural Information Processing Systems*, pp. 12060–12070, 2019.

[28] J. Fliege, L. G. Drummond, and B. F. Svaiter, "Newton's method for multiobjective optimization," *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 602–626, 2009.

[29] J. Fliege and H. Xu, "Stochastic multiobjective optimization: sample average approximation and applications," *Journal of optimization theory and applications*, vol. 151, no. 1, pp. 135–162, 2011.

[30] J. Fliege, A. I. F. Vaz, and L. N. Vicente, "Complexity of gradient descent for multiobjective optimization," *Optimization Methods and Software*, vol. 34, no. 5, pp. 949–959, 2019.

[31] J.-A. Désidéri, "Multiple-gradient descent algorithm (MGDA) for multiobjective optimization," *Comptes Rendus Mathematique*, vol. 350, no. 5-6, pp. 313–318, 2012.

[32] J.-A. Désidéri, "Multiple-gradient descent algorithm for pareto-front identification," in *Modeling, Simulation and Optimization for Science and Technology*, pp. 41–58, Springer, 2014.

[33] Q. Mercier, F. Poirion, and J.-A. Désidéri, "A stochastic multiple gradient descent algorithm," *European Journal of Operational Research*, vol. 271, no. 3, pp. 808–817, 2018.

[34] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems*, pp. 527–538, 2018.

[35] A. Katrutsa, D. Merkulov, N. Tursynbek, and I. Oseledets, "Follow the bisector: a simple method for multi-objective optimization," *arXiv preprint arXiv:2007.06937*, 2020.

[36] M. Suteu and Y. Guo, "Regularizing deep multi-task networks using orthogonal gradients," *arXiv preprint arXiv:1912.06844*, 2019.

[37] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," *arXiv preprint arXiv:2001.06782*, 2020.

[38] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1871–1880, 2019.

[39] M. Long, Z. Cao, J. Wang, and S. Y. Philip, "Learning multiple tasks with multilinear relationship networks," in *Advances in neural information processing systems*, pp. 1594–1603, 2017.

[40] Y. Yang and T. Hospedales, "Deep multi-task representation learning: A tensor factorisation approach," *arXiv preprint arXiv:1605.06391*, 2016.

[41] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3994–4003, 2016.

[42] C. Rosenbaum, T. Klinger, and M. Riemer, "Routing networks: Adaptive selection of non-linear functions for multi-task learning," *arXiv preprint arXiv:1711.01239*, 2017.

[43] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4822–4829, 2019.

[44] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, "Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5334–5343, 2017.

[45] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?," *arXiv preprint arXiv:1905.07553*, 2019.

[46] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, "Efficient parametrization of multi-domain deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8119–8127, 2018.

[47] E. Meyerson and R. Miikkulainen, "Beyond shared hierarchies: Deep multitask learning through soft layer ordering," *arXiv preprint arXiv:1711.00108*, 2017.

[48] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, "Attentive single-tasking of multiple tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1860, 2019.

[49] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3712–3722, 2018.

[50] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.