Быстрая оптимизация мультизадачных моделей

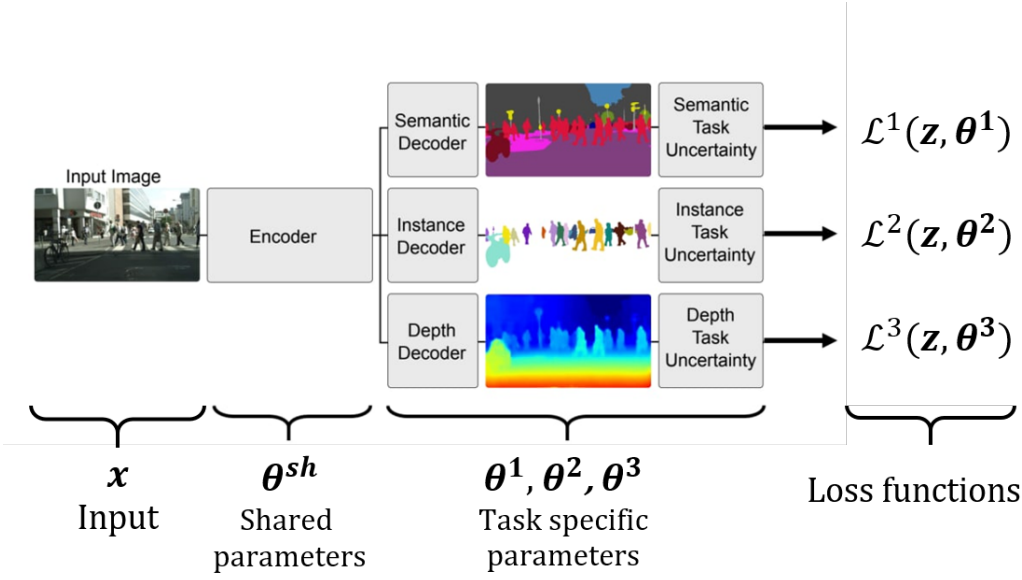
Филатов Андрей

Московский физико-технический институт
Кафедра интеллектуальных систем

Научный руководитель: д.ф.-м.н. Стрижов В.В.
Консультант: Меркулов Д.М.

23 Июня, 2021

Мультизадачные модели



Многокритериальная оптимизация

Заданы T функции потерь (задач) $\mathscr{L}^t$. Требуется их одновременная оптимизации:

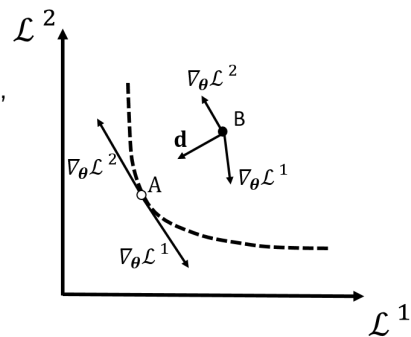
$$\min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \mathsf{L}\left(\boldsymbol{\theta}^{sh},\boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T\right) = \min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \left(\mathscr{L}^1\left(\boldsymbol{\theta}^{sh},\boldsymbol{\theta}^1\right),\ldots,\mathscr{L}^T\left(\boldsymbol{\theta}^{sh},\boldsymbol{\theta}^T\right)\right)^\top,$$

где $\boldsymbol{\theta}^{sh}$ — параметры общие параметры, а $\boldsymbol{\theta}^t$ — отдельные параметры для каждой задачи

Парето станционарность

Пусть функции $\mathscr{L}^t$ — непрерывно-дифференцируемы. Тогда точка $\boldsymbol{\theta} = [\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}^T]$ — Парето станционарная, если существует набор коэффициентов $\alpha^1, \ldots, \alpha^T \geq 0$,

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



Методы решения задачи многокритериальной оптимизации

1. Взвешивание[1]: сводим многокритериальную оптимизацию к однокритериальной оптимизации следующим образом:

$$\mathscr{L}(\boldsymbol{\theta}) = \sum_{t=1}^{T} w_t \mathscr{L}^t(\boldsymbol{\theta}).$$

2. Методы нулевого порядка: эволюционные алгоритмы и многокритериальной байесовская оптимизация.[2]
3. Градиентные методы[3]: градиентный спуск, метод Ньютона.

---

[1] Chen и др., "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks".

[2] Deb и др., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II".

[3] Fliege и Svaiter, "Steepest descent methods for multicriteria optimization".

Градиентные методы

- Multiple-gradient descent algorithm (MGDA)[4]

$$\min_{\alpha^1,\dots,\alpha^T} \left\{ \left\| \sum_{t=1}^{T} \alpha^t \nabla_{\boldsymbol{\theta}} \mathscr{L}^t(\boldsymbol{\theta}) \right\|_2^2 \mid \sum_{t=1}^{T} \alpha^t = 1, \alpha^t \geq 0 \quad \forall t \right\},$$

$$\boldsymbol{d}^* = \sum_{t=1}^{T} \tilde{\alpha}^t \nabla_{\boldsymbol{\theta}} \mathscr{L}^t.$$

- Min-max подход[5]

$$\min_{\boldsymbol{d}} \max_{t} \left(\frac{\partial \mathscr{L}^t}{\partial \boldsymbol{\theta}} \boldsymbol{d} \right)_t + g^t(\boldsymbol{d})$$

Рассматривая $g^t(\boldsymbol{d}) = \|\boldsymbol{d}\|^2$ получаем двойственную к MGDA.
Рассматривая $g^t(\boldsymbol{d}) = \frac{1}{2} \boldsymbol{d}^T \boldsymbol{H}^t \boldsymbol{d}$ получаем метод Ньютона ($\boldsymbol{H}^t$ гессиан $\mathscr{L}^t$).

[4] Désidéri, "Mgda variants for multi-objective optimization".

[5] Fliege и Svaiter, "Steepest descent methods for multicriteria optimization".

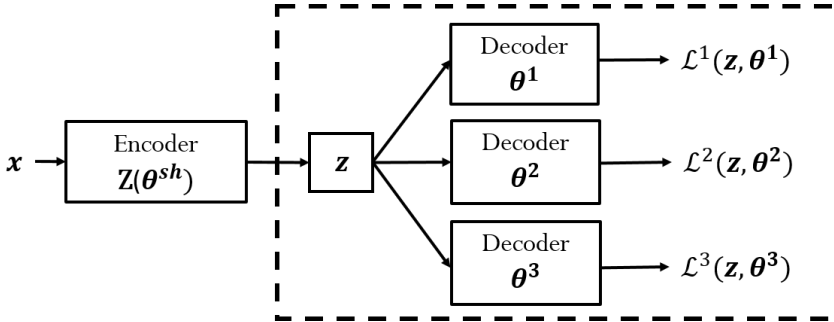
Мотивация

- Для градиентных методов оптимизации мультизадачных моделей теоретически обосновано лишь использование линейного поиска для нахождение шага.
- Линейный поиск неэффективен на практике из высокой вычислительной стоимости.

Цель

Создать вычислительно эффективный метод линейного поиска.

Быстрый линейный поиск



В алгоритме быстрого линейного поиска мы оптимизируем в скрытом пространстве Z.

Алгоритм линейного поиска

Пусть получено $\boldsymbol{d}$ — направление убывания всех функций: $\forall t\ \nabla_{\boldsymbol{\theta}} \mathscr{L}^t \boldsymbol{d} < 0$.
Необходимо найти шаг η, чтобы:

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

Теорема (Fliege 2000)[6]

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

[6] Fliege и Svaiter, "Steepest descent methods for multicriteria optimization".

Алгоритм линейного поиска

Правило Армихо

На каждом шаге градиентного спуска нам найти шаг η, чтобы выполнялось следующее правило Армихо $\forall t \in \{1 \dots T\}$:

$$\mathscr{L}^t(\boldsymbol{\theta}^{sh} - \eta \boldsymbol{d}_{sh}, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathscr{L}^t) \leq \mathscr{L}^t - \eta\beta \left\| \frac{\partial \mathscr{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 - \eta\beta \left(\frac{\partial \mathscr{L}^t}{\partial \boldsymbol{\theta}^{sh}} \right)^\top \boldsymbol{d}_{sh}.$$

Модифицированное правило Армихо

На каждом шаге градиентного спуска нам найти шаг η, чтобы выполнялось следующее правило Армихо $\forall t \in \{1 \dots T\}$:

$$\mathscr{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathscr{L}^t) \leq \mathscr{L}^t - \eta\beta \left\| \frac{\partial \mathscr{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 - \eta\beta \left(\frac{\partial \mathscr{L}^t}{\partial \boldsymbol{z}} \right)^\top \boldsymbol{d}_z.$$

Основной результат

Модифицированное правило Армихо

На каждом шаге градиентного спуска нам найти шаг η, чтобы выполнялось следующее правило Армихо :

$$\mathscr{L}^t(\boldsymbol{z} - \eta \boldsymbol{d}_z, \boldsymbol{\theta}^t - \eta \nabla_{\boldsymbol{\theta}^t} \mathscr{L}^t) \leq \mathscr{L}^t - \eta\beta \left\| \frac{\partial \mathscr{L}^t}{\partial \boldsymbol{\theta}^t} \right\|^2 - \eta\beta \left(\frac{\partial \mathscr{L}^t}{\partial \boldsymbol{z}} \right)^\top \boldsymbol{d}_z$$

Теорема (Филатов, 2021)

Предел последовательности, созданной градиентным спуском с модифицированным правилом Армихо, является Парето станционарной точкой.

Базовый[7] и быстрый алгоритмы

Algorithm 1: Backtracking

Require: β, γ, lr_{ub}
Ensure: Learning rate $\eta = lr_{ub}/\gamma$

1: **repeat**
2: $\eta \leftarrow \gamma \cdot \eta$
3: $\tilde{\boldsymbol{\theta}}^{sh} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot \boldsymbol{d}_{sh}$
4: **for** $t \leftarrow 1$ to T **do**
5: $\tilde{\boldsymbol{\theta}}^{t} \leftarrow \boldsymbol{\theta}^{t} - \eta \cdot \nabla_{\boldsymbol{\theta}^t} \mathscr{L}^t$
6: **end for**
7: **until** правило Армихо
8: **for** $t \leftarrow 1$ to T **do**
9: $\boldsymbol{\theta}^{t}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{t}$
10: **end for**
11: $\boldsymbol{\theta}^{sh}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{sh}$

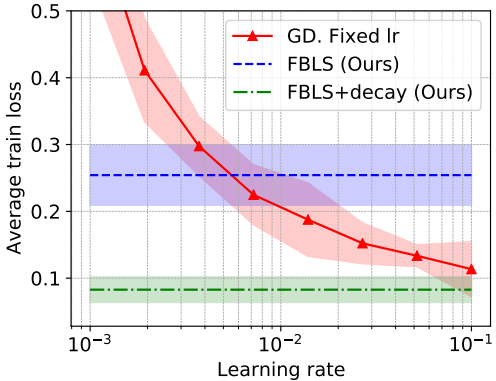
Algorithm 2: Fast backtracking (Ours)

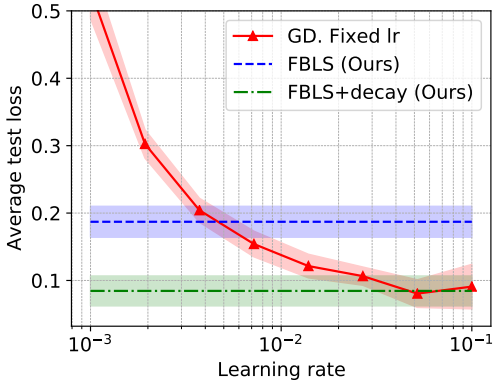
Require: β, γ, lr_{ub}
Ensure: Learning rate $\eta = lr_{ub}/\gamma$

1: **repeat**
2: $\eta \leftarrow \gamma \cdot \eta$
3: $\boldsymbol{z} \leftarrow \boldsymbol{z} - \eta \cdot \boldsymbol{d}_{z}$
4: **for** $t \leftarrow 1$ to T **do**
5: $\tilde{\boldsymbol{\theta}}^{t} \leftarrow \boldsymbol{\theta}^{t} - \eta \cdot \nabla_{\boldsymbol{\theta}^t} \mathscr{L}^t$
6: **end for**
7: **until** правило Армихо
8: **for** $t \leftarrow 1$ to T **do**
9: $\boldsymbol{\theta}^{t}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{t}$
10: **end for**
11: $\boldsymbol{\theta}^{sh}_{new} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot \frac{\partial \boldsymbol{\theta}^{sh}}{\partial \boldsymbol{z}} \boldsymbol{d}_{z}$

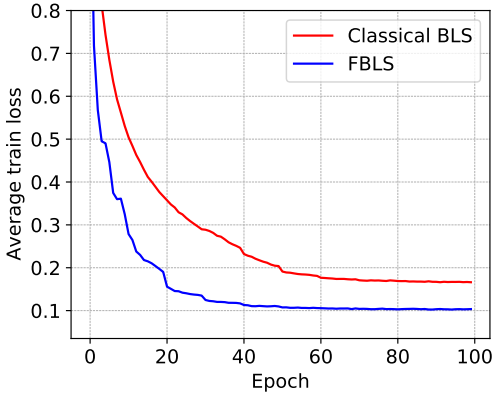
[7] Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives".

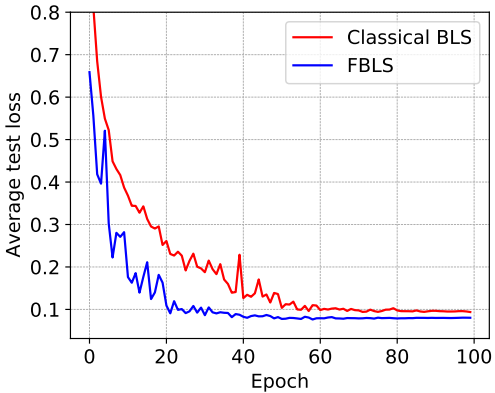
Сравнение предложенного метода с градиентным спуском на MultiMNIST



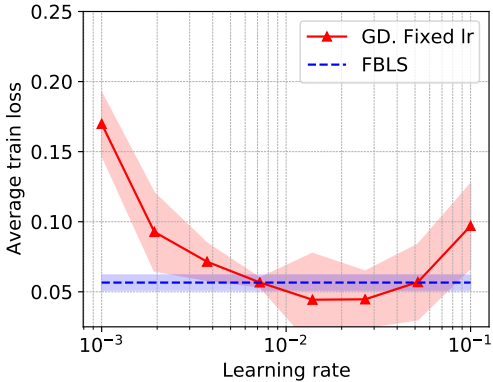


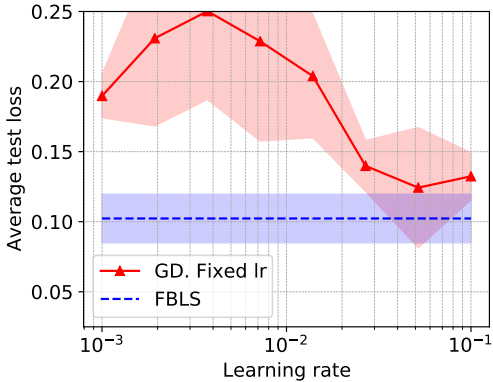
Сравнение предложенного метода с базовым линейным поиском на MultiMNIST





Сравнение предложенного метода с градиентным спуском на CIFAR-10





Сравнение времени работы алгоритмов

	MNIST ↓	CIFAR-10 ↓	Cityscapes ↓
FBLS (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[8]	**0.95** (136)	**0.14** (80)	-

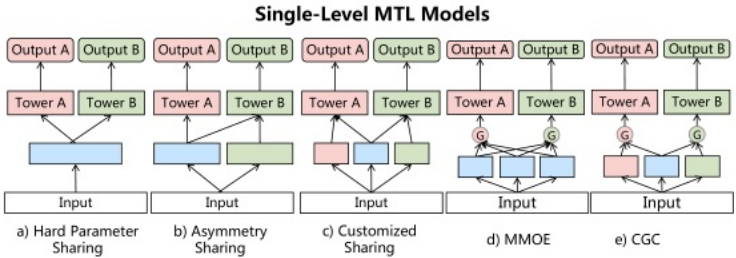
По результатам экспериментов было получено, что быстрый линейный поиск эффективней обычного линейного поиска и незначительно увеличивает временные затраты по сравнению с градиентным спуском.

[8] Sener и Koltun, "Multi-task learning as multi-objective optimization".

Результаты, выносимые на защиту

1. Предложен алгоритм быстрого линейного поиска для мультизадачных моделей.
2. Подтверждена теоретическая сходимость быстрого линейного поиска к Парето станционарной точке.
3. Проверена практическая эффективность метода на задачах MultiMNIST, CIFAR-10, Cityscapes.

Примеры различных моделей



Парето доминирование

Считаем, что точка $\boldsymbol{\theta}_1$ доминируется точкой $\boldsymbol{\theta}_2$ если:

$$\forall t \in \{1, \ldots, T\}\ \mathscr{L}^t(\boldsymbol{\theta}_2) \leq \mathscr{L}^t(\boldsymbol{\theta}_1)$$

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

Парето оптимальность

Точка $\boldsymbol{\theta} = [\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t]$ — Парето оптимальная тогда и только тогда, когда не существует другой точки $\hat{\boldsymbol{\theta}}$, которая Парето доминирует $\boldsymbol{\theta}$.

Парето станционарность — необходимое условие Парето оптимальности.