

Быстрая оптимизация мультизадачных моделей

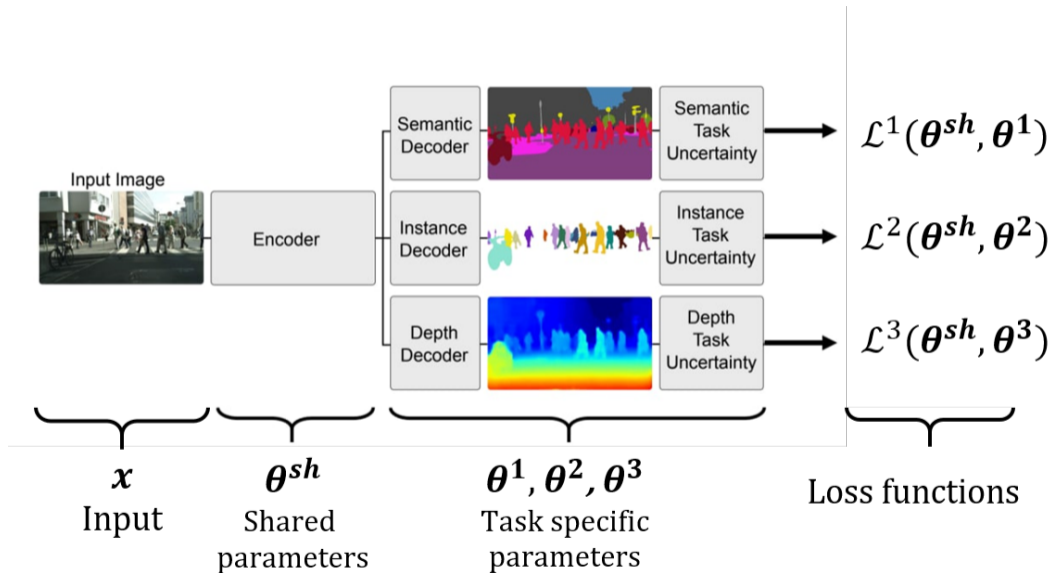
Филатов Андрей

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) \mathcal{L}^t . Требуется их одновременная оптимизация:

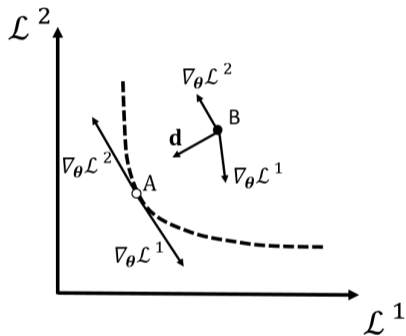
$$\min_{\substack{\boldsymbol{\theta}^{sh}, \\ \boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T}} \mathbf{L}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T) = \min_{\substack{\boldsymbol{\theta}^{sh}, \\ \boldsymbol{\theta}^1, \dots, \boldsymbol{\theta}^T}} \left(\mathcal{L}^1(\boldsymbol{\theta}^{sh},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

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

Парето стационарность

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

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



Методы решения задачи многокритериальной оптимизации

1. Взвешивание¹: сводим многокритериальную оптимизацию к однокритериальной оптимизации следующим образом:

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

2. Методы нулевого порядка: эволюционные алгоритмы и многокритериальной байесовская оптимизация.²
3. Градиентные методы³: градиентный спуск, метод Ньютона.

¹Chen и др., “Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks”.

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

³Fliege и Svaiter, “Steepest descent methods for multicriteria optimization”.

Градиентные методы

- ▶ Multiple-gradient descent algorithm (MGDA)⁴

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

$$\mathbf{d}^* = \sum_{t=1}^T \tilde{\alpha}^t \nabla_{\boldsymbol{\theta}^{sh}} \mathcal{L}^t(\boldsymbol{\theta}).$$

- ▶ Min-max подход⁵

$$\min_{\mathbf{d}} \max_t \left(\frac{\partial \mathcal{L}^t}{\partial \boldsymbol{\theta}^{sh}} \mathbf{d} \right)_t + g^t(\mathbf{d})$$

Рассматривая $g^t(\mathbf{d}) = \|\mathbf{d}\|^2$ получаем двойственную к MGDA.

Рассматривая $g^t(\mathbf{d}) = \frac{1}{2} \mathbf{d}^T \mathbf{H}^t \mathbf{d}$ получаем метод Ньютона (\mathbf{H}^t гессиан \mathcal{L}^t).

⁴Désidéri, “Mgda variants for multi-objective optimization”.

⁵Fliege и Svaiter, “Steepest descent methods for multicriteria optimization”.

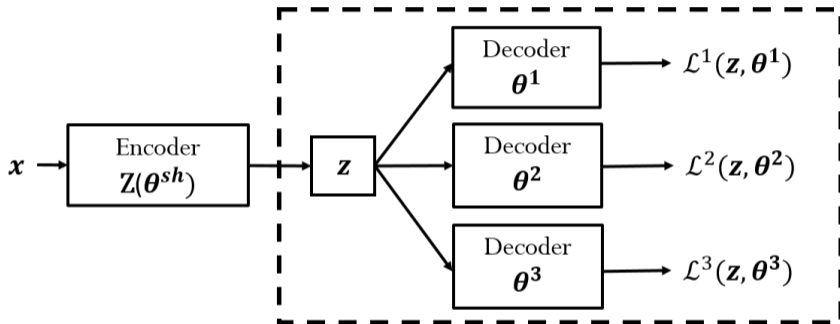
Мотивация

- ▶ Для градиентных методов оптимизации мультизадачных моделей теоретически обосновано лишь использование линейного поиска для нахождения 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 \mathbf{d} — направление убывания всех функций: $\forall t \nabla_{\boldsymbol{\theta}} \mathcal{L}^t \mathbf{d} < 0$.
Необходимо найти шаг η , чтобы:

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

Теорема (Fliege 2000)⁶

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

⁶Fliege 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 $\forall t \in \{1 \dots T\}$:

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

Модифицированное правило Армихо

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

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

Основной результат

Модифицированное правило Армихо

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

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

Теорема (Филатов, 2021)

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

Algorithm 1: Backtracking

Require: β, γ, lr_{ub}

Ensure: Learning rate $\eta = lr_{ub}/\gamma$

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

Algorithm 2: Fast backtracking (Ours)

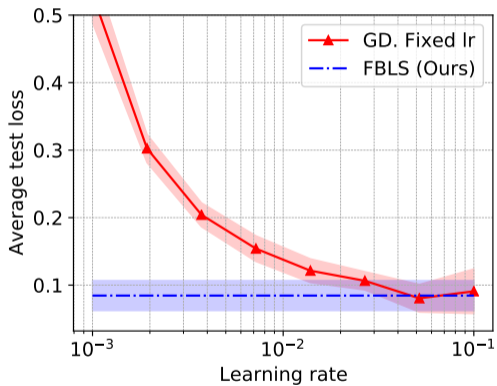
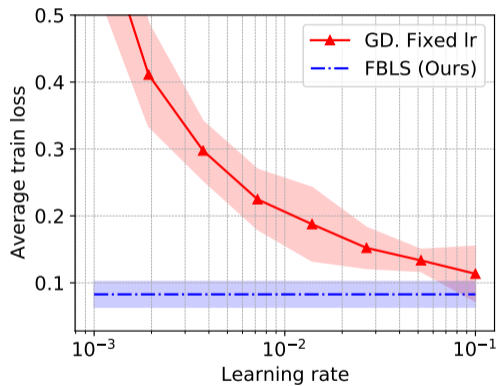
Require: β, γ, lr_{ub}

Ensure: Learning rate $\eta = lr_{ub}/\gamma$

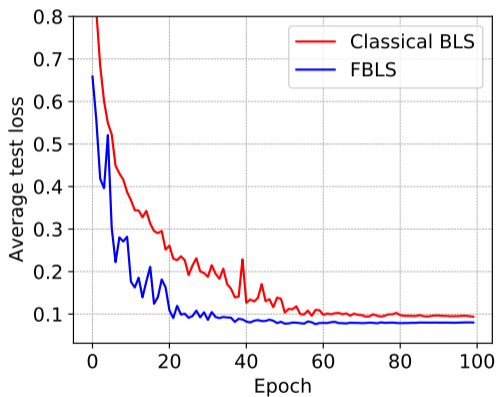
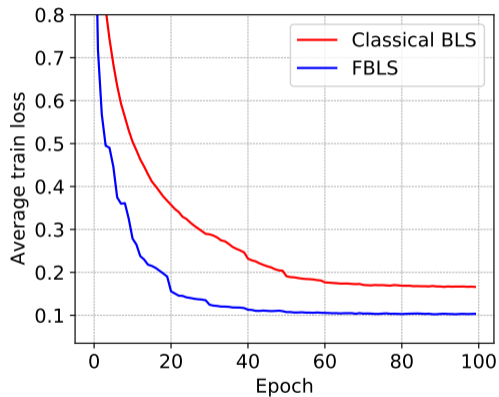
- 1: repeat
 - 2: $\eta \leftarrow \gamma \cdot \eta$
 - 3: $z \leftarrow z - \eta \cdot d_z$
 - 4: for $t \leftarrow 1$ to T do
 - 5: $\tilde{\theta}^t \leftarrow \theta^t - \eta \cdot \nabla_{\theta^t} \mathcal{L}^t$
 - 6: end for
 - 7: until правило Армихо
 - 8: for $t \leftarrow 1$ to T do
 - 9: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

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

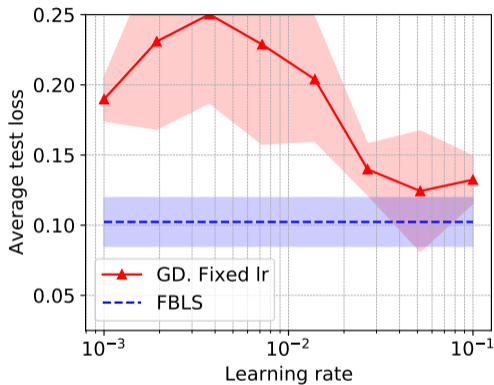
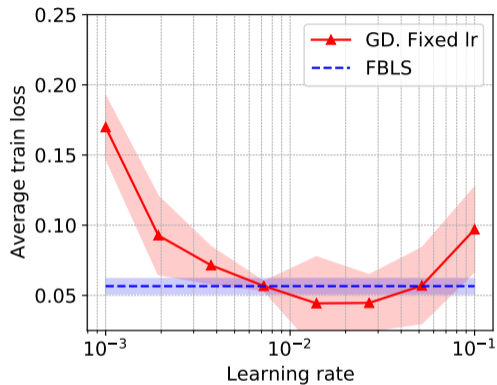
Сравнение предложенного метода с градиентным спуском на MultiMNIST



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-UB⁸	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.

⁸Sener и Koltun, “Multi-task learning as multi-objective optimization”.

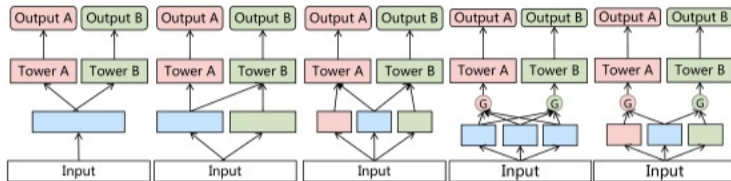
[†]В скобках указано число секунд на одну эпоху.

Результаты, выносимые на защиту

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

Примеры различных моделей

Single-Level MTL Models



a) Hard Parameter Sharing

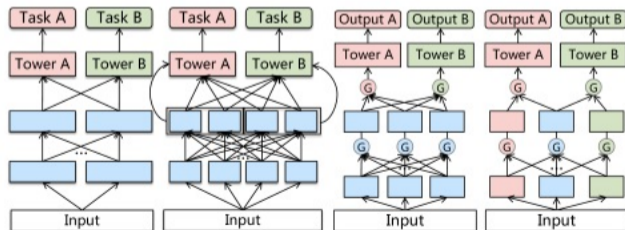
b) Asymmetry Sharing

c) Customized Sharing

d) MMOE

e) CGC

Multi-Level MTL Models



f) Cross-Stitch Network

g) Sluice Network

h) ML-MMOE

i) PLE

Парето доминирование

Считаем, что точка θ_1 доминируется точкой θ_2 если:

$$\forall t \in \{1, \dots, T\} \mathcal{L}^t(\theta_2) \leq \mathcal{L}^t(\theta_1)$$

$$\exists i : \mathcal{L}^i(\theta_2) < \mathcal{L}^i(\theta_1)$$

Парето оптимальность

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

Парето стационарность — необходимое условие Парето оптимальности.