«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

Содержание

Аннотация

Мультизадачное обучение - это эффективный метод совместного решения нескольких задач путем обучения надежного представляния. Оптимизация модели мультизадачного обучения является более сложной задачей, чем решение одной задача, так как задачи могут противоречить друг другу. Исходя из теоретических результатов, сходимость к оптимальной точке гарантирована, если размер шага выбирается с помощью линейного поиска, чтобы удовлетворить условию Армихо. Но, как правило, линейный поиск размера шага не является лучшим выбором из-за больших вычислительных затрат. В данной работе предлагается новая идея для алгоритмов линейного поиска в мультизадачном обучении, которая использует структурные свойства мультизадачных моделей. Идея была проверена на линейного поиска типа "backtracking". Проведено сравнение предложенного алгоритма с классическим бэктрекингом 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 , и сохраняет конкурентоспособное вычислительное время и производительность по сравнению с градиентным методом с постоянной шагом.

1 Введение

Многозадачное обучение (MTL) [5] - это подход к индуктивному переносу, который улучшает обобщение за счет использования информации об области, содержащейся в обучающих сигналах связанных задач, в качестве индуктивного предубеждения. Это достигается за счет параллельного обучения задач с использованием общего представления; то, что изучено для каждой задачи, может помочь лучше изучить другие задачи. Многие подходы MTL показали свою эффективность и высокую производительность во многих областях, таких как компьютерное зрение [38, 7], обработка естественного языка [64, 9], и распознавание речи [29].

Нейронные сети - это современный метод для решения различных задач машинного обучения, и многозадачное обучение не является исключением. Оптимизация нейронных сетей часто использует методы градиентного спуска. Было предложено несколько методов адаптации градиентного спуска для решения многозадачной задачи. Первый подход - скаляризация [32], когда многозадачная задача сводится к однозадачной. Классическим методом является взвешивание — рассмотрение многозадачной проблемы как взвешенной комбинации однозадачных проблем. Этот подход сильно зависит от правильного выбора веса. Адаптивное взвешивание [6] [35] методы введены для решения этой проблемы. Второй подход - это метод min-max [23]. В этом случае мы ищем направление, минимизирующее все функции. Первоначально двойственная задача минимизации нормы в технике min-max появилась в [17] и представила новый взгляд на проблему.

Классические методы адаптивного размера шага, такие как Adam [36], NAG [54], RMSProp [66], могут быть применены для многозадачного обучения и хорошо работают на практике, но не имеют теоретических гарантий сходимости. Использование методов линейного поиска, таких как обратный путь [2, 71, 69, 68] золотой поиск и параболическая интерполяция имеет теоретические гарантии сходимости, но требует дополнительных вызовов функций, что в случае нейронной сети имеет большую стоимость. Особенно это непрактично при многозадачном обучении, когда есть несколько модулей, специфичных для конкретной задачи, и каждый вызов функции требует прямого шага через огромный кодер.

Основываясь на структурных свойствах многозадачных моделей с "жесткой общей параметризацией мы предлагаем новую идею оптимизации. Наша идея заключается в том, чтобы выполнять только проходы вперед по модели, специфичной для

конкретной задачи, что значительно сокращает затраты времени на поиск в строке, поскольку кодер, содержащий большую часть затрат, неактивен. Мы протестировали наш метод на Multi-MNIST [60], CIFAR-10 и Cityscapes [10] с быстрым бэктрекингом против классического бэктрекинга и стандартного градиентного спуска. По сравнению с классическим бэктрекингом мы получаем более быструю сходимость и более точное решение. По сравнению с классическими подходами с фиксированной скоростью обучения мы получаем конкурентоспособную производительность без исчерпывающего поиска гиперпараметров и увеличение времени только на 15%.

Далее мы кратко рассмотрим исследовательские работы, связанные с данной статьей. статье. Затем, в разделе 3 предложенная модель ресайзера обсуждается в деталях. В разделе 4 представлены наши результаты, и, наконец, в разделе 5 мы делаем вывод.

Введение объясняет мотивацию работы: где возникает данная задача, почему её решение так важно, как её решали до сих пор, в чём недостатки этих решений, и что нового предлагает автор. Введение лучше писать напоследок, так как в ходе работы обычно происходит переосмысление постановки задачи.

Вводятся на неформальном уровне основные понятия, необходимые для понимания постановки задачи. Определяются цели исследования и формулируется постановка задачи.

Во введении можно привести краткий анализ источников информации. Однако если литературный обзор большой, ему лучше посвятить отдельный раздел.

В конце введения даётся краткое содержание работы по разделам, при этом отмечается, какие подходы, методы, алгоритмы предлагаются автором впервые. Перечисляются основные результаты и 1–2 самых важных вывода работы.

Введение может быть близко по содержанию к тексту доклада на защите.

2 Обзорно-постановочный раздел работы

Заголовки разделов и подразделов в данном документе приводятся для примера и должны быть заменены на более содержательные, отражающие суть работы. Можно оставить «Введение» и «Заключение».

2.1 Основные понятия и определения

Вводятся общепринятые понятия и обозначения со ссылками на литературу.

2.2 Обзор современного состояния проблемы

Излагаются известные результаты со ссылками на литературу. Необходимые для дальнейшего известные факты могут формулироваться как теоремы без доказательств, но со ссылками на источники.

Определение 2.1. *Математический текст хорошо структурирован, если в нём выделены определения, теоремы, утверждения, примеры, и т. д., а неформальные рассуждения (мотивации, интерпретации) выделены в отдельные подпараграфы (для этого хорошо подходит команда* `\paragraph{`*текст*`}`*).*

Теорема 2.1. *Не менее* 90% *коллег, заинтересовавшихся Вашей работой, прочитают в ней не более* 10% *текста, причём это будут именно те разделы, которые не содержат формул.*

Следствие 2.1.1. *Надо уделять большое внимание не только формальному изложению, но и неформальным мотивациям и интерпретациям результатов. Иначе основные идеи работы невозможно будет понять быстро.*

Следствие 2.1.2. *Основные идеи Вашего текста должны оставаться в целом понятными, если читать его, пропуская все формулы.*

Замечание 2.1. Здесь показано применение окружений Def, Theorem, Corollary, Remark.

2.3 Формальная постановка задачи

К данному моменту читатель ознакомлен со всеми необходимыми понятиями и подготовлен к точным формулировкам задач, решаемых автором в данной работе.

3 Основной раздел работы

Алгоритм 3.1. Backtracking(γ, lr_{ub})

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

повторять
  $\eta \leftarrow \gamma \cdot \eta$
  $\boxed{\tilde{\boldsymbol{\theta}}^{sh} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot d_{sh}}$
  для $t \leftarrow 1$ to T
    $\tilde{\boldsymbol{\theta}}^{t} \leftarrow \boldsymbol{\theta}^{t} - \eta \cdot \nabla_{\boldsymbol{\theta}^t} L^t$
пока Armijo condition
для $t \leftarrow 1$ to T
  $\boldsymbol{\theta}^{t}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{t}$
$\boxed{\boldsymbol{\theta}^{sh}_{new} \leftarrow \tilde{\boldsymbol{\theta}}^{sh}}$

Алгоритм 3.2. Fast backtracking(γ, lr_{ub})

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

повторять
  $\eta \leftarrow \gamma \cdot \eta$
  $\boxed{z \leftarrow z - \eta \cdot d_z}$
  для $t \leftarrow 1$ to T
    $\tilde{\boldsymbol{\theta}}^{t} \leftarrow \boldsymbol{\theta}^{t} - \eta \cdot \nabla_{\boldsymbol{\theta}^t} L^t$
пока Armijo condition
для $t \leftarrow 1$ to T
  $\boldsymbol{\theta}^{t}_{new} \leftarrow \tilde{\theta}^{t}$
$\boxed{\boldsymbol{\theta}^{sh}_{new} \leftarrow \boldsymbol{\theta}^{sh} - \eta \cdot \frac{\partial \boldsymbol{\theta}^{sh}}{\partial z} d_z}$

Графические иллюстрации могут быть подготовлены в любом графическом формате, в частности, `BMP`, `PNG` или `EPS`.

Желательно, чтобы рисунки были чёрно-белыми или grayscale (оттенки серого). При чёрно-белой печати передача цвета плохо предсказуема. Жёлтый цвет, как правило, не виден. Синий, зелёный и красный могут оказаться неотличимыми.

Рисунки вставляются с помощью окружения `figure` и могут разрывать текст в любом месте. Положением плавающих рисунков на странице управляет необяза-

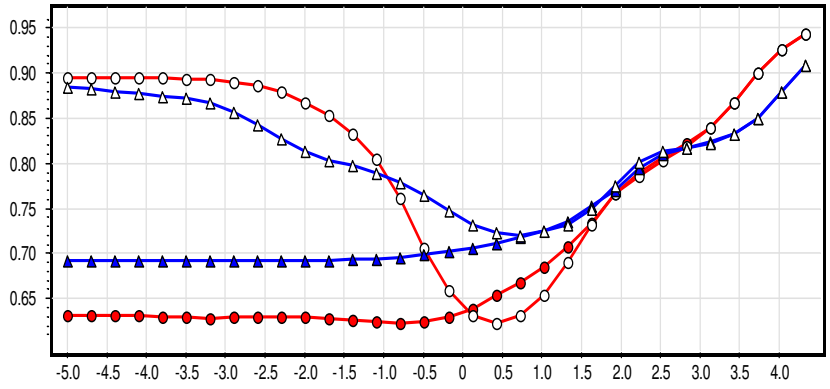


Рис. 1: Подпись должна размещаться под рисунком. ВНИМАНИЕ! Красные и синие линии при чёрно-белой печати будут выглядеть как чёрные.

тельный аргумент команды `\begin{figure}`. Подпись делается *под рисунком* командой `\caption`, см. рис. **??**.

Оси на графике должны быть подписаны. Если на графике несколько кривых, обязательно должна быть легенда. В подписи под графиком обычно пишут, что за зависимость изображена, и при каких условиях эксперимента был получен данный график.

Для фотографических изображений лучше подходит формат `JPEG`, для растровых — `PNG`. Рекомендуется тот и другой сначала перевести в `EPS` (Encapsulated PostScript). Это делается с помощью утилиты `bmeps`, входящей в пакет MiKTEX:

```
bmeps -c -t jpg myfig.jpeg myfig.eps
bmeps -c -t png myfig.png myfig.eps
```

Для векторной графики лучше найти способ записать изображение непосредственно в формате `EPS`. Например, в MATLAB имеется функция сохранения графика в `EPS`. Если же делать скриншот экрана и записывать изображение через растровый формат, качество изображения будет скверным. Преобразование файла растрового изображения в `EPS` не делает его «настоящим» векторным, просто каждый пиксел рисуется прямоугольничком.

Алгоритмы оформляются в стиле псевдокода с помощью окружения `algorithmic`, внутри которого определены стандартные ключевые слова `\IF`, `\FOR`, `\WHILE`, и др., которые при печати дают, соответственно, **если**, **для**, **пока**, и т. д. Шаги алгоритма нумеруются автоматически, и на них можно ссылаться, см. шаг 5 алгоритма 3.3.

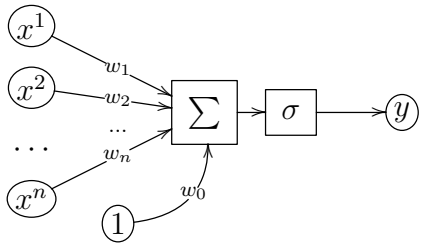
Алгоритм 3.3. Показаны все допустимые управляющие конструкции.

Вход: x, y;

Выход: $z = F(x, y)$;

```
1: инициализация: b := a;
2: для i = 1, ..., n
3:    для всех w ∈ W таких, что w ⩾ 0
4:       повторять
5:          важный шаг: вычисление вектора u_i;
6:       пока ‖u_i − u_{i−1}‖ > ε;
7: если a > 0 то
8:    пока W ≠ ∅
9:       W := W − {a};
10: иначе если a = 0 то
11:    цикл    — бесконечный цикл
12:       при определённых условиях
13:       выход;
14: иначе    — при a < 0
15:    a := 1;
```

Рисование графов с помощью окружения `network` из пакета XY-pic. В стиле `Diplo.sty` определены две вспомогательные команды. Команда `\nnNode` задаёт имя и координаты вершины, команда `\nnLink` связывает две ранее поименованных вершины. Внешний вид вершин и связей задаётся средствами пакета XY-pic:



Некоторые правила типографики. Скобки всех видов набираются вплотную к тексту, который они окружают. Знаки препинания набираются слитно с предшествующим текстом и отдельно от последующего.

Кавычки делаются знаками меньше–больше: <<*текст*>>. Использовать в роли кавычек символ " нельзя!

Многоточия в тексте и формулах делаются командой `\dots`.

Тире делается командой `"---` и отделяется от предшествующего и последующего текста пробелами: `Знание␣"---␣сила`.

В длинных словах с дефисом, таких, как «счётно-аддитивно», дефис делается командой `"=`, иначе слово не будет переноситься: `счётно"=аддитивно`. Команда-дефис `"~` запрещает переносы: F-преобразование, `$F$"~пре\-образование`.

Неразрывный пробел `~` ставится между коротким предлогом и последующим словом, а также между очень короткой формулой и связанным с ней по смыслу словом: `число~$N$ в~$k$~раз больше, чем~$n$`.

Между идущими подряд формулами лучше вставлять дополнительный пробел:

`$a=1,b=2$`	$a=1,b=2$ — плохо
`$a=1$, $b=2$`	$a=1$, $b=2$ — получше
`$a=1$,\: $b=2$`	$a=1,\ b=2$ — хорошо
`$a=1$,\; $b=2$`	$a=1,\ b=2$ — хорошо

В русскоязычной математической литературе принято при переносе формулы на новую строку дублировать на новой строке знак математического оператора. В стилевом файле `Diplo.sty` для этого определена команда `\brop`. Пример: $\frac{1}{2}\sqrt{b^2-4ac} = {}$ $= \sqrt{\left(\frac{b}{2}\right)^2 - ac}$.

Иногда в формуле надо убрать пробелы вокруг знака операции. Например, если знак $\times$ используется не как произведение, а для указания размеров матрицы или растрового изображения, то он не должен окружаться пробелами:

`$640\times 480$`	640×480 — плохо
`$640{\times}480$`	$640{\times}480$ — хорошо

Дополнительный пробел `\quad` рекомендуется вставлять между выражениями, идущими через запятую в выключной формуле.

Короткий пробел `\,` ставится в инициалах и сокращениях:

`т. е., и т. д.`	т. е., и т. д. — плохо
`т.\,е., и~т.\,д.`	т.\,е., и т.\,д. — хорошо
`Ю.И.Журавлёв`	Ю.И.Журавлёв — плохо
`Ю.\,И.\,Журавлёв`	Ю.\,И.\,Журавлёв — хорошо

Не желательно использовать жирный шрифт для выделения *важных слов* или *терминов*. Это делается командой `\emph{`*текст*`}`.

Разумное форматирование исходного кода заметно облегчает работу с текстом для Вас и научного руководителя. По возможности придерживайтесь нескольких простых правил:

- начинайте каждое предложение с новой строки;
- команды `\begin`, `\end`, `$$`, `\[`, `\]`, `\section`, `\subsection`, `\paragraph` `\item`, `\bibitem`, `\par`, `\label` набирайте отдельной строкой;
- внутритекстовые формулы, за исключением совсем коротких, набирайте отдельной строкой;
- описания длинных формул разбивайте на строки; используйте форматирование исходного текста с отступами, набирая отдельной строкой команды скобок `\left`, `\right`, и т. п., как показано в Примере 3.1.

Пример 3.1. Без «правильного» форматирования было бы легко запутаться в скобках и похожих частях формулы:

$$R'_N(F) = \frac{1}{N}\sum_{i=1}^{N}\Big(P(+1\,|\,x_i)C\big(+1, F(x_i)\big) + P(-1\,|\,x_i)C\big(-1, F(x_i)\big)\Big).$$

```
\begin{align*}
   R'_N(F)
   = \frac1N \sum_{i=1}^N
   \Bigl( &
      P(+1\cond x_i)C\bigl(+1,F(x_i)\bigr)+{}
   \\ &
      P(-1\cond x_i)C\bigl(-1,F(x_i)\bigr)
   \Bigr).
\end{align*}
```

4 Вычислительные эксперименты

Описывается прикладная задача, параметры анализируемых данных (например, сколько объектов, сколько признаков, каких они типов), параметры эксперимента

(например, как генерировались модельные данные, как производился скользящий контроль). Результаты экспериментов представляются в виде таблиц и графиков. Объясняется точный смысл всех обозначений на графиках, строк и столбцов в таблицах. Приводятся выводы: в какой степени результаты экспериментов согласуются с теорией? Достигнут ли желаемый результат? Обнаружены ли какие-либо факты, не нашедшие объяснения, и которые нельзя списать на «грязный» эксперимент?

Цель данного раздела: продемонстрировать, что предложенная теория работает на практике; показать границы её применимости; рассказать о новых экспериментальных фактах.

Чисто теоретические работы могут не содержать данного раздела (на практике не работает, ну и не надо — зато теория красивая). Кстати, теоретики имеют право не догадываться, где, кому и когда их теории пригодятся.

5 Заключение

Основные результаты работы

- Предложен метод оптимизации мультизадачных моделей;
- Подтверждена теоретическая сходимость предложенного метода;
- Проведены вычислительные эксперименты, которые подтвердили эффективность метода на прикладных задачах в различных условиях.

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 времени уменьшения и насколько уменьшать. Поэтому 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 Andreani, JM Martınez, M Salvatierra, and F Yano. Quasi-newton methods for order-value optimization and value-at-risk calculations. *Pacific Journal of Optimization*, 2:11–33, 2006.

[2] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[3] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[4] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018.

[7] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. *arXiv preprint arXiv:1901.05808*, 2019.

[8] Giorgio Chiandussi, Marco Codegone, Simone Ferrero, and Federico Erminio Varesio. Comparison of multi-objective optimization methodologies for engineering applications. *Computers & Mathematics with Applications*, 63(5):912–942, 2012.

[9] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167, 2008.

[10] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

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

[12] Yunfei Cui, Zhiqiang Geng, Qunxiong Zhu, and Yongming Han. Multi-objective optimization methods and application in energy saving. *Energy*, 125:681–704, 2017.

[13] Andrea Dal Pozzolo, Giacomo Boracchi, Olivier Caelen, Cesare Alippi, and Gianluca Bontempi. Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE transactions on neural networks and learning systems*, 29(8):3784–3797, 2017.

[14] Indraneel Das and John E Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[15] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM journal on optimization*, 8(3):631–657, 1998.

[16] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International conference on parallel problem solving from nature*, pages 849–858. Springer, 2000.

[17] Jean-Antoine Désidéri. Mgda variants for multi-objective optimization. 2012.

[18] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5-6):313–318, 2012.

[19] Jean-Antoine Désidéri. Multiple-gradient descent algorithm for pareto-front identification. In *Modeling, Simulation and Optimization for Science and Technology*, pages 41–58. Springer, 2014.

[20] Yezid Donoso and Ramon Fabregat. *Multi-objective optimization in computer networks using metaheuristics*. CRC Press, 2016.

[21] LM Grana Drummond and Benar F Svaiter. A steepest descent method for vector optimization. *Journal of computational and applied mathematics*, 175(2):395–414, 2005.

[22] Joerg Fliege, LM Grana Drummond, and Benar Fux Svaiter. Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.

[23] Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51(3):479–494, 2000.

[24] Jörg Fliege, A Ismael F Vaz, and Luís Nunes Vicente. Complexity of gradient descent for multiobjective optimization. *Optimization Methods and Software*, 34(5):949–959, 2019.

[25] Jörg Fliege and Huifu Xu. Stochastic multiobjective optimization: sample average approximation and applications. *Journal of optimization theory and applications*, 151(1):135–162, 2011.

[26] A Ghane-Kanafi and E Khorram. A new scalarization method for finding the efficient frontier in non-convex multi-objective problems. *Applied Mathematical Modelling*, 39(23-24):7483–7498, 2015.

[27] Ting Gong, Tyler Lee, Cory Stephenson, Venkata Renduchintala, Suchismita Padhy, Anthony Ndirango, Gokce Keskin, and Oguz H Elibol. A comparison of loss weighting strategies for multi task learning in deep neural networks. *IEEE Access*, 7:141627–141632, 2019.

[28] Han Xu Yao Ma Hao-Chen, Liu Debayan Deb, Hui Liu Ji-Liang Tang Anil, and K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020.

[29] Zhen Huang, Jinyu Li, Sabato Marco Siniscalchi, I-Fan Chen, Ji Wu, and Chin-Hui Lee. Rapid adaptation for deep neural networks through multi-task learning. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[30] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th international conference on machine learning*, pages 427–435, 2013.

[31] Yaochu Jin. *Multi-objective machine learning*, volume 16. Springer Science & Business Media, 2006.

[32] Jahn Johannes. Scalarization in vector optimization. *Mathematical Programming*, 29(2):203–218, 1984.

[33] Alexandr Katrutsa, Daniil Merkulov, Nurislam Tursynbek, and Ivan Oseledets. Follow the bisector: a simple method for multi-objective optimization. *arXiv preprint arXiv:2007.06937*, 2020.

[34] AM Katrutsa and VV Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.

[35] Alex Kendall, Yarin Gal, and Roberto 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*, pages 7482–7491, 2018.

[36] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[37] Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[38] Iasonas 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*, pages 6129–6138, 2017.

[39] Isabelle Leang, Ganesh Sistu, Fabian Burger, Andrei Bursuc, and Senthil Yogamani. Dynamic task weighting methods for multi-task networks in autonomous driving systems. *arXiv preprint arXiv:2001.02223*, 2020.

[40] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, pages 12060–12070, 2019.

[43] Chunming Liu, Xin Xu, and Dewen Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398, 2014.

[44] Shikun Liu, Edward Johns, and Andrew J Davison. End-to-end multi-task learning with attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.

[45] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. Learning multiple tasks with multilinear relationship networks. In *Advances in neural information processing systems*, pages 1594–1603, 2017.

[46] Yongxi Lu, Abhishek Kumar, Shuangfei Zhai, Yu Cheng, Tara Javidi, and Rogerio 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*, pages 5334–5343, 2017.

[47] Kevis-Kokitsi Maninis, Ilija Radosavovic, and Iasonas Kokkinos. Attentive single-tasking of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1860, 2019.

[48] R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[49] Quentin Mercier, Fabrice Poirion, and Jean-Antoine Désidéri. A stochastic multiple gradient descent algorithm. *European Journal of Operational Research*, 271(3):808–817, 2018.

[50] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.

[51] Elliot Meyerson and Risto Miikkulainen. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. *arXiv preprint arXiv:1711.00108*, 2017.

[52] Kaisa Miettinen and Marko M Mäkelä. On scalarizing functions in multiobjective optimization. *OR spectrum*, 24(2):193–213, 2002.

[53] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.

[54] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[55] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[56] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018.

[57] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *arXiv preprint arXiv:1711.01239*, 2017.

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

[59] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829, 2019.

[60] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[61] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018.

[62] Paolo Soda. A multi-objective optimisation approach for class imbalance learning. *Pattern Recognition*, 44(8):1801–1810, 2011.

[63] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *arXiv preprint arXiv:1905.07553*, 2019.

[64] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*, 2018.

[65] Mihai Suteu and Yike Guo. Regularizing deep multi-task networks using orthogonal gradients. *arXiv preprint arXiv:1912.06844*, 2019.

[66] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[67] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

[68] Sharan Vaswani, Frederik Kunstner, Issam Laradji, Si Yi Meng, Mark Schmidt, and Simon Lacoste-Julien. Adaptive gradient methods converge faster with over-parameterization (and you can do a line-search). *arXiv preprint arXiv:2006.06835*, 2020.

[69] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *Advances in Neural Information Processing Systems*, pages 3732–3745, 2019.

[70] Douglas AG Vieira, Ricardo HC Takahashi, and Rodney R Saldanha. Multicriteria optimization with a multiobjective golden section line search. *Mathematical Programming*, 131(1-2):131–161, 2012.

[71] Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2):226–235, 1969.

[72] Bing Xue, Mengjie Zhang, and Will N Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2012.

[73] Yongxin Yang and Timothy Hospedales. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*, 2016.

[74] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[75] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11197–11206, 2020.

[76] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[77] Adrien Zerbinati, Jean-Antoine Desideri, and Régis Duvigneau. Comparison between mgda and paes for multi-objective optimization. 2011.

[78] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.