Multiple Constraints and Non-regular Solution in Deep Declarative Network

Suikei Wang

A thesis submitted for the degree of
Master of Machine Learning and Computer Vision
The Australian National University

October 2020

Except where otherwise indicated, this thesis is my own original work.

Suikei Wang
18 October 2020

to my parents, for their unconditional love

Acknowledgments

The past two years at the Australian National University have been an invaluable experience for me. When I started my Master of Machine Learning and Computer Vision at the beginning of 2019, I could barely understand lectures, knew little about the country, and had never heard of the term "convex optimization". It is unbelievable that I have been doing a research project on this topic for a whole year. ANU has the top tier research group in this realm and how honored I am to be a postgraduate student here.

First and foremost, I would like to express my deep and sincere gratitude to my supervisor, Stephen Gould. I knew Stephen before the admission of the program and how privileged I am to work and learn with one of the most brilliant minds in our field. He always has an insightful and high-level view on this topic. More importantly, Stephen is extremely kind and patient as he is always willing to discuss and share ideas with us during the weekly meetings. Although sometimes I am stuck in some problems he used to give me constructive suggestions.

I would like to thank Dylan Campbell and Miaomiao Liu – another two giants of the Australian Centre for Robotic Vision – for offering my supervision and being my second examiner on my thesis. Dylan is always enthusiasm and knowledgable, and his ideas and works inspire me a lot in my research. Although I did not work with Miaomiao before, her works in computer vision area are very impressive, which make me work hardly to complete this research. I want to thank her for her time exaiming my thesis seminar.

During my research, I have very fun collaboration and discussion with collaborators in our research group: Jiteng Ma, Jingyi Zhang, Yuchong Yao, Ruikai Cui, and Wenbo Du. Although we almost never meet in person due to the COVID-19, we still share many ideas related to our research through the social network. I greatly enjoyed the teamwork with them.

Outside the research group, I am very lucky to be surrounded by many great friends. Yao Liu, Ce Wang and Nuocheng Tian – We almost study together every single day during the academic year, even in the public holiday. I am always impressed by their insightful ideas and I very appreciate the brainstorm with them. Mingrui Zhao – As my closest partner, a charming Engineering & Mathematics double degree undergraduate student and my math "teacher", he always encourages me to pursue my academic ambition and I learned a lot from him.

I thank my parents: Shiwu Wang and Hongyin Shen. As traditional Asian parents, they never restrict my life planning, and always encourage me to fulfill my dream. I hope that they will a little proud of me.

Abstract

Put your abstract here.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

1.2 Thesis Outline

Following the two central themes we mentioned above, this thesis consists of two parts – PART I Deep Declarative Network: Multiple Constrained Declarative Nodes and Part II Deep Declarative Nodes: Non-regular Solution.

PART I focus on the regular points of multiple constrained declarative nodes in deep declarative network, with some basic examples of different constraints nodes.

In Chapter 2, we first give an overview of the theory of optimization, with the discussion of the optimality. Next we formally define the unconstrained, equality constrained and inequality constrained problems. We then discuss several related works on the solutions of these problems. Finally, differentiable neural network, an application of numerical optimization in machine learning is briefly discussed with various modern deep learning algorithms based on it.

In Chapter 3, we present the details of the deep declarative network. We begin with the overview structure and how it works through the specific declarative nodes. We then introduce the learning progress of the deep declarative network. We analysize the backpropagation through the declarative nodes in different constrained problems. Finally, we give multiple examples of the implementation of the deep declarative nodes in both equality constrained and inequality constrained optimization problems. This chapter is based on works of Gould et al. [2019].

In Chapter 4, we point out the limitation of current deep declarative nodes and address some ideas for the future improvements in the optimization process. We also look to the practical application of the deep declarative network in computer vision tasks, which will be forseeable.

PART II is an extension of the deep declarative nodes in non-regular solution problems, which cannot be solved through the traditional numerical optimization methods. Detailedly, In Chapter 5, we give an overview of the non-regular solution problems. We list the general non-regular solution cases: Overdetermined system, rand deficiency and non-convex feasible set. We also introduce various related works for solving the problems when the gradient result is not a regular point.

In Chapter 6, we demonstrate various possible solutions for each non-regular point case. Since for non-regular solution problems, there is no exact solution, we can only appxiate the closed result. Thus, we also compare and discuss the results of each method on minimizing the final loss.

We will finally conclude in Chapter 7. Proofs for the important theorems and definitions are given in the Appendix.

1.3 Contribution

Part I

Deep Declarative Network: Multiple Constrained Declarative Nodes

Chapter 2

An Overview of Numerical Optimization

In this chapter, we aim to provide readers with an overview of numerical optimization. We begin with the theory of optimization (Section 2.1), from the existence of optimizers, to the optimality conditions for both unconstrained and constrained problems with duality. As the theoretical background of optimization, this field provides a solid solution for the algorithm.

We then formally define the optimization of unconstrained and constrained problems in Section 2.2 and describe the general regular solution for these problems based on the gradient calculation.

Next, we discuss briefly the differentiable optimization in the neural network, which is a novel end-to-end network structure involving optimization problems in each layer with and without constraints. (Section 2.3). Finally, we give a summary of the numerical optimization for solving unconstrained and constrained problems in Section 2.4.

2.1 Theory of Optimization

2.1.1 Existence of Optimizers

In optimization, a basic question is to determine the existence of a global minimizer for a given function f. There are several sufficient conditions on f to guarantee the existence, and the optimizer falls in the feasible set of solutions. For a feasible set, some related definitions are following:

Definition 2.1. A subset $\Omega \in \mathbb{R}^n$ is called

- *bounded* if there is a constant $R > 0$ such that $\|x\| \leq R$ for all $x \in \Omega$
- *closed* if the limit point of any convergent sequence in Ω always lies in Ω
- *compact* if any sequence $\{x_k\}$ in Ω contains a subsequence that converges to a point in Ω

The following result gives a characterization of compact sets in $\mathbb{R}$. When we find the minimum or maximum solution for the problem, there exists a lower bound or upper bound but not necessarily an optimal solution. Therefore, we have some additional requirements.

Firstly, we give the definition of compact sets in Lemma 2.2. [Oman, 2017] gives a brief proof.

Lemma 2.2 (Bolzano-Weierstrass theorem)**.** A subset Ω in $\mathbb{R}^n$ is *compact* if and only if it is bounded and closed.

We also assume that the function f is continuous and "$+\infty$ at infinity". More precisely, $f(x) \to +\infty$ if $|x| \to +\infty$. Such a function is called *inf-compact* or *coercive*. [Nocedal and Wright, 2006] Then the problem can be restricted to a bounded set and existence of a global minimum x^* is guaranteed: a continuous function has a minimum on a compact set. This theorem is defined as follows and the proof is given in Appendix A.1.1.

Theorem 2.3. *[Nocedal and Wright, 2006] If f is a continuous function defined on a compact set Ω in $\mathbb{R}$, then f has a global minimizer x^* on Ω i.e. there exists $x^* \in \Omega$ such that $f(x^*) \leq f(x)$ for all $x \in \Omega$*

More general, based on the definition of coercive function f, we can give following theorem. Proof is given in Appendix A.1.2.

Theorem 2.4. *[Nocedal and Wright, 2006] If $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous coercive function, then f has at least one global minimizer.*

Theorem 2.4 requires the continuity of f which is somewhat restrictive for applications. However, we can replace it by the lower semi-continuity of f which is a rather weaker condition.

Definition 2.5. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{\pm\infty\}$. Then f is called *lower semi-continuous* at a point $x_0 \in \mathbb{R}^n$ if for any sequence $f(x_k)$ converging to x_0 here holds $f(x_0) \leq \lim_{k\to\infty} f(x_k)$. f is called *lower semi-continuous* if f is lower semi-continuous at every point.

Recall our assumptions on function f, it is a continuous function, which is always lower semi-continuous. However, lower semi-continuous functions are not necessarily continuous. For instance, a binary function equals to 0 when $x \leq 0$ and equals to 1 when $x > 0$ is not continuous at $x_0 = 0$. However, since it is greater than 0 for all x and $f(0) = 0$, we have $f(0) = 0 \leq \liminf_{x\to 0} f(x)$ and it is lower semi-continuous at $x_0 = 0$.

The theorem of the existence of the optimizer of lower semi-continuous function is given as follows and the proof is given in Appendix A.1.3

Theorem 2.6. *[Nocedal and Wright, 2006] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a lower semi-continuous function. If f has a nonempty, compact sublevel set $D := \{x \in \mathbb{R}^n : f(x) \leq \alpha\}$, then f achieves a global minimizer on $\mathbb{R}$*

Also, we introduce the definition of convex function and convex set which are important in regular optimization problems.

Definition 2.7. A function f is convex when

$$f(\alpha x+(1-\alpha)y) \leq \alpha f(x)+(1-\alpha)f(y) \quad \text{for all } x,y, \text{ and } \alpha \in]0,1[$$

A set $C \subset \mathbb{R}^n$ is convex when

$$\alpha x+(1-\alpha)y \in C \quad \text{for all } x,y \text{ in } C, \text{ and } \alpha \in]0,1[$$

The problem we are going to discuss in this part is convex and regular, which means its gradient can be computed and the solution exists. However, although the existence of the optimizer is sufficient, for different problems, the optimality conditions are different. In the next two sections, we will give necessary and sufficient conditions for both unconstrained and constrained problems.

2.1.2 Optimality Conditions for Unconstrained Problems

Firstly, we consider the unconstrained minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(x), \tag{2.1}$$

where f is given function on $\mathbb{R}^n$.

In order to determine the minimizer, it is important to understand what can happen at a minimizer, and at what condition a point must be a minimizer. Now we have to recognize the optimum point. There are two necessary conditions and one sufficient condition given below [Nocedal and Wright, 2006]. The proof is given in Appendix A.1.4.

Theorem 2.8. *Necessary and Sufficient Conditions. Let $f : \Omega \to \mathbb{R}$ be a funcion defined on a set $\Omega \subset \mathbb{R}^n$ and let x^* be an interior point of Ω that is a local minimizer of f.*
Necessary conditions:

- *(NC1) If f is differentiable at x^*, then x^* is a critical point of f, i.e. $\nabla f(x^*) = 0$.*
- *(NC2) If f is twice continuous differentiable on Ω, then the Hessian $\nabla^2 f(x^*)$ is positive semidefinite.*

Sufficient condition (SC1): if x^* is such that $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite, then x^* is a local minimum. (i.e. $f(x) \geq f(x^*)$ for x close to x^*)

Any point satisfying (NC1) as the minimizer of f is called a *critical* or *stationary* point of f. In the objective function f is convex, (NC1) is also the sufficient condition for the global minimum of the solution.

Let us see an example of unconstrained minimization problem. Supposed we have to determine the minimization of function

$$f(x,y) = x^4 - 4xy + y^4$$

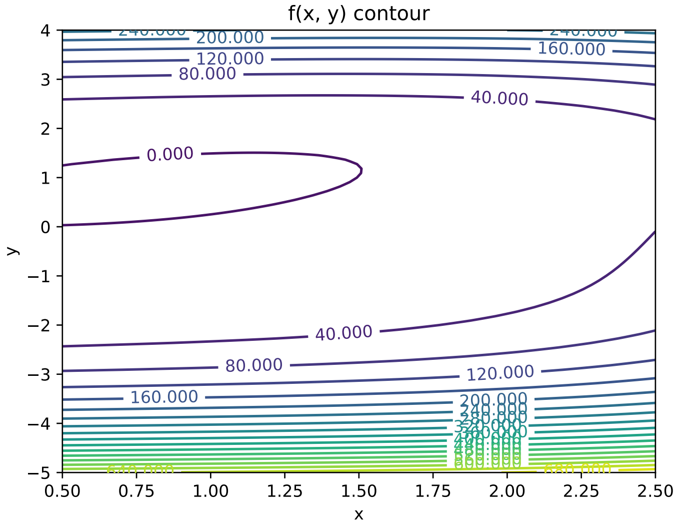


Figure 2.1: Contour Graph of $f(x,y) = x^4 - 4xy + y^4$

From the definition of function f, it is clear that f is continuous. Then we can expand f by writing

$$f(x,y) = \left(x^4 + y^4\right)\left(1 - \frac{4xy}{x^4 + y^4}\right)$$

we can see f is coercive. Also, we give the contour graph of function f in Figure 2.1. Therefore f has global minimizers which are critical points. Now according to (NC1), we can find the global minimizer through solving the derivative of f equaling to zero:

$$0 = \nabla f(x,y) = \begin{pmatrix} 4x^3 - 4y \\ -4x + 4y^3 \end{pmatrix}$$

Thus, $y = x^3$ and $x = y^3$. Consequently $y = y^9$, i.e.

$$0 = y - y^9 = y\left(1 - y^8\right) = y\left(1 - y^4\right)\left(1 + y^4\right) = y(1-y)(1+y)\left(1 + y^2\right)\left(1 + y^4\right)$$

This implies $y = 0, 1, -1$. Thus f has three critical points $(0,0), (1,1), (-1,-1)$. Then we can evaluate f as these points since they may be local minimizer:

$$f(0,0) = 0, \quad f(1,1) = -2, \quad f(-1,-1) = -2$$

It achieves the same global minimum value on $(1,1)$ and $(-1,-1)$. Therefore, they are both global minimizers of f. Figure 2.2 shows the function $f(x,y)$ at these two optimal points.

From this example, we verify that through (NC1), we can find the global minimizer. However, not all continuous functions with critical points have any maximizer

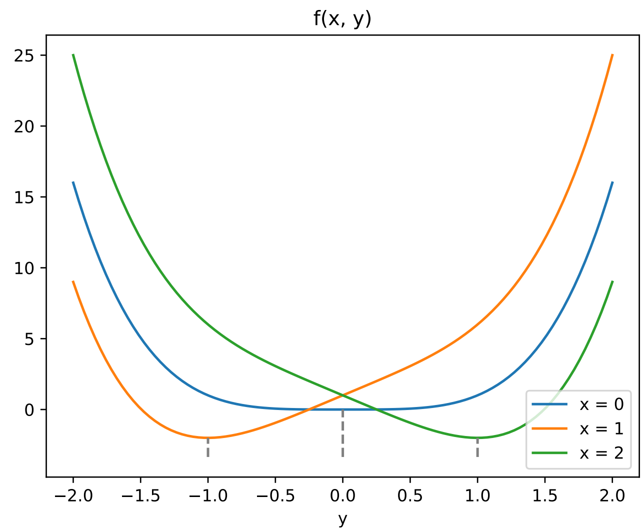


Figure 2.2: Function $f(x,y)$ at $x = 0$, $x = -1$ and $x = 1$

or minimizer. If the function goes to infinity along its axes or a line, it does not have any maximizer or minimizer although it has a critical point. The condition of the minimizer as the critical point is that the function f should be a convex function with continuous first partial derivatives.

Let us move to the sufficient condition (SC1). The result obtained under this theorem is best possible for general functions. Specifically, for a convex function f is defined on a convex set $\Omega \subset \mathbb{R}^n$, any local minimizer of f is also a global minimizer. Moreover, if a function f is strictly convex, it has at most one global minimizer.

2.1.3 Optimality Conditions for Constrained Problems

A general formulation for constrained optimization problems is as follows:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } \begin{cases} c_i(x) = 0 & \text{for } i = 1, \cdots, m_e, \\ c_i(x) \le 0 & \text{for } i = m_e + 1, \cdots, m \end{cases} \end{aligned} \tag{2.2}$$

where f and c_i are smooth real-valued functions on $\mathbb{R}^n$, and m_e and m are nonnegative integers with $m_e < m$. We set

$$\mathscr{E} := \{1, \cdots, m_e\} \quad \text{and} \quad \mathscr{I} := \{m_e + 1, \cdots, m\}$$

as index sets of equality constraints and inequality constraints, respectively.

Here, f is so-called the objective function, and $c_i, i \in \mathscr{E}$ and $\mathscr{I}$ are equality constraints and inequality constraints respectively.

To solve the optimization problem (2.2), we define the feasible set of it to be

$$\mathscr{F} := \{x \in \mathbb{R}^n : c_i(x) = 0 \text{ for } i \in \mathscr{E} \text{ and } c_i(x) \leq 0 \text{ for } i \in \mathscr{I}\}$$

Any point $x \in \mathscr{F}$ is called a feasible point of (2.2) and we call (2.2) infeasible if $\mathscr{F} = 0$. Also, in this feasible set, a feasible point $x^* \in \mathscr{F}$ is called a local minimizer of (2.2) if it is the minimum solution in a neighborhood (strict local minimizer if it is the only one minimum solution). The definition of the global minimizer and strict global minimizer is similar, whose neighborhood is the whole feasible set.

Let us move to the constraints in this problem. For equality constraints, they are strictly equivalent. However, for inequality constraints, there are some exceptions. Let x^* be a local minimizer of (2.2). If there is an index $i \in \mathscr{I}$ such that $c_i(x^*) < 0$, then, x^* is still the local minimizer of the problem obtained by deleting i-th constraint. In this situation, we say that the i-th constraint is inactive at x^* since it does not have any effect on the solution. A general definition of active and inactive inequality constraints is as follows:

Definition 2.9. At a feasible point $x \in \mathscr{F}$, the index $i \in \mathscr{I}$ is said to be *active* if $(x) = 0$ and *inactive* if $c_i(x) < 0$.

In the next chapter, we will give different processes for different cases of active or inactive inequality constraints in the deep declarative nodes. In this chapter, we only focus on the necessary and sufficient conditions for a feasible point x to be a local minimizer of (2.2). These conditions will be derived by considering the change of f on the feasible set along with certain directions. We give the lemma for the condition of local minimizer $x^* \in \mathscr{F}$ as follows, which can be proved through Taylor's formula in Appendix A.1.5.

Lemma 2.10. If $x^* \in \mathscr{F}$ is a local minimizer of (2.2), then

$$d^T \nabla f(x^*) \geq 0 \quad \text{for all } d \in T_{x^*}\mathscr{F}$$

where $T_{x^*}\mathscr{F}$ is the set of all vectors tangent to $\mathscr{F}$.

However, we may not be able to extract useful results from this lemma, since $T_{x^*}\mathscr{F}$ depends only on the geometry of $\mathscr{F}$ but not on the constraints functions c_i. Not all local minimum falls on the boundary of the constraint function, which is a part of $T_{x^*}\mathscr{F}$. Therefore, it is necessary to introduce linearized feasible directions to give a characterization of $T_{x^*}\mathscr{F}$ in terms of c_i.

Definition 2.11. Given $x \in \mathscr{F}$, we define

$$\mathrm{LFD}(x) := \left\{d \in \mathbb{R}^n : d^T \nabla c_i(x) = 0 \text{ for } i \in \mathscr{E}; d^T \nabla c_i(x) \leq 0 \text{ for } i \in \mathscr{I} \cap \mathscr{A}(x)\right\}$$

and call it the set of linearized feasible directions of $\mathscr{F}$ at x.

Heuristically, for $i \in \mathscr{E}$ we should travel along directions d with $d^T \nabla c_i(x) = 0$ in order to stay on the curve $c_i(x) = 0$; for $i \in \mathscr{I}$ we should travel along directions with

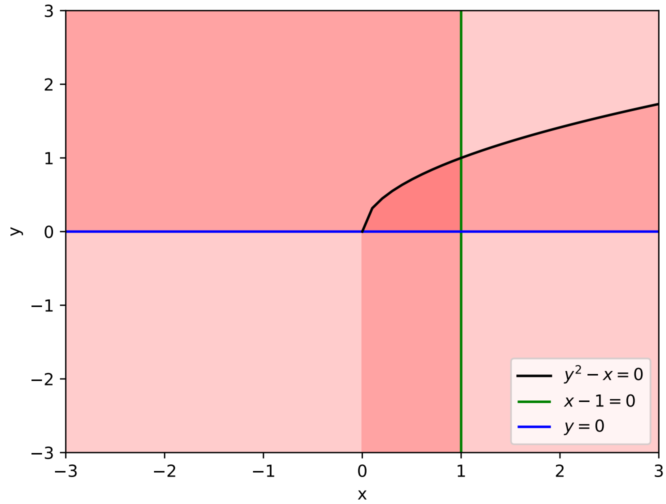


Figure 2.3: Feasible set of constraints c_1, c_2 and c_3

$d^T \nabla c_i(x) \leq 0$ in order to stay in the region $c_i(x) \leq 0$. Let us see an example of the linearized feasible directions and the tangent. Supposed we are considering a set $\mathscr{F}$ with variables $(x, y) \in \mathbb{R}^2$ and three inequality constraints functions:

$$
\begin{aligned}
c_1(x, y) &= x - 1 \leq 0 \\
c_2(x, y) &= -y \leq 0 \\
c_3(x, y) &= y^2 - x \leq 0
\end{aligned}
$$

We can illustrate the feasible set of constriants c_1, c_2 and c_3 in Fig 2.3. The active set of $0 = (0, 0)$ is $\{2, 3\}$, since $c_1(0) = -1 < 0$, which is inactive. And we can get the derivative of c_2 and c_3 at 0:

$$\nabla c_2(0) = (0, -1)^T \quad \text{and} \quad \nabla c_3(0) = (-1, 0)^T$$

Then we have the linearized feasible directions on $x = 0$:

$$
\begin{aligned}
\text{LFD}(0) &= \left\{ d \in \mathbb{R}^2 : d^T \nabla c_2(0) \leq 0 \text{ and } d^T \nabla c_3(0) \leq 0 \right\} \\
&= \left\{ d \in \mathbb{R}^2 : d \geq 0 \right\}
\end{aligned}
$$

which equals to the set of all vectors tangent to the feasible set $T_0 \mathscr{F}$.

Unlike the unconstrained optimization problem, the first order necessary condition of the existence of the optimizer is different since we should consider its linearized feasible directions and constraints feasibility. This is so-called the Karush-Kuhn-Tucker theorem:

Theorem 2.12 (Karush-Kuhn-Tucker Theorem)**.** *Let $x^* \in \mathscr{F}$ be a local minimizer of* *problem (2.2). If*

$$T_{x^*}\mathscr{F} = \mathrm{LFD}(x^*),$$

then there exists $\lambda^* = (\lambda_1^*, \cdots, \lambda_m^*)^T \in \mathbb{R}^m$ *such that*

$$\nabla f(x^*) + \sum_{i \in \mathscr{E} \cup \mathscr{I}} \lambda_i^* \nabla c_i(x^*) = 0, \quad (\textit{Lagrangian stationary})$$

$$\left.\begin{array}{ll} c_i(x^*) = 0 & \textit{for all } i \in \mathscr{E}, \\ c_i(x^*) \leq 0 & \textit{for all } i \in \mathscr{I}, \end{array}\right\} \quad (\textit{primal feasibility})$$

$$\lambda_i^* \geq 0 \quad \textit{for alli} \in \mathscr{I}, \quad (\textit{dual feasibility})$$

$$\lambda_i^* c_i(x^*) = 0 \quad \textit{for alli} \in \mathscr{E} \cup \mathscr{I}. \quad (\textit{complementary slackness})$$

This set of equations are Karush-Kuhn-Tucker (KKT) conditions and a point x^* *is called a KKT point if there exists* λ^* *such that* (x^*, λ^*) *satisfies the KKT conditions.*

For constrained optimization problem, the classic solution is using Lagrange multipliers [Bertsekas, 2014]. This introduces the function

$$\mathscr{L}(x, \lambda) := f(x) + \sum_{i \in \mathscr{E} \cup \mathscr{I}} \lambda_i c_i(x)$$

which is called the Lagrange function. x is the primal variables and $\lambda_i, i = 1, \ldots, m$ are the Lagrange multipliers or the dual variables. According to the Lagrange multipliers method, we can solve this problem through the gradient of the Lagrange function:

$$\nabla_x \mathscr{L}(x, \lambda) = \nabla f(x) + \sum_{i \in \mathscr{E} \cup \mathscr{I}} \lambda_i \nabla c_i(x)$$

Therefore, the first equation in KKT conditions can be written as

$$\nabla_x \mathscr{L}(x^*, \lambda^*) = 0$$

2.2 Solution of Unconstrained and Constrained Optimization Problems

According to the sufficient conditions for unconstrained optimization problems, we can easily compute the optimality through the first and second derivative of the objective function. For equality and inequality constrained problems, the introduction of Lagrangian $\mathcal{L}$ is useful for their closed-form solution. Gould et al. [2016] collected both argmin and argmax bi-level optimization results with and without constraints, which also provide insightful examples of these cases. Amos and Kolter [2017] also present a solution for exact, constrained optimization within a neural network. In this thesis, we only focus on argmin problems, but the argmax problems have similar results.

In this section, we are going to provide some background for the solution of both unconstrained and constrained optimization problems, which is based on the gradient of the regular point.

2.2.1 Unconstrained Optimization

For unconstrained optimization problems, the solution is easy to obtain since we only need to focus on the optimality of the objective function. We consider an objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$:

$$y(x) \in \operatorname{argmin} f(x, y)$$

The derivative of $y(x)$ with respect to x is

$$\frac{dy(x)}{dx} = -[\frac{\partial^2 f}{\partial y(x)^2}]^{-1} \frac{\partial^2 f}{\partial x \partial y(x)} \tag{2.3}$$

which can be proved through differentiating and chain rule. [A.2.1]

A very classic example of the unconstrained minimization problem based on a closed convex nonempty set is the L2 norm $\|\cdot\|_2$. Let $\Omega \in \mathbb{R}^n$ be a closed convex nonempty set. For any $x \in \mathbb{R}^n$, the minimization problem is defined as follows:

$$\min_{y \in \Omega} \|y - x\|_2^2$$

This problem has a unique minimizer, which can be denoted by $P_\Omega(x)$, the Euclidean projection of x onto Ω.

Proof. Let $m := \inf_{y \in \Omega} \|y - x\|_2^2$. Since $\Omega \neq \emptyset$, we have $0 \leq m < \infty$. Let $\{y_k\} \subset \Omega$ be a minimizing sequence such that $\|y_k - x\|_2^2 \to m$ as $k \to \infty$. Thus $\|y_k - x\|_2^2 \leq m + 1$ for large k which implies that $\|y_k\|_2 \leq \|x\|_2 + \sqrt{m+1}$ for large k. Therefore $\{y_k\}$ is a bounded sequence. Consequently $\{y_k\}$ has a convegent subsequence $\{y_{k_l}\}$ with limit y^*. Since Ω is closed, we have $y^* \in \Omega$, Thus

$$m = \lim_{l \to \infty} \left\|y_{k_l} - x\right\|_2^2 = \|y^* - x\|_2^2$$

which means that m is achieved at y^*, i.e. the given minimization problem has a solution.

Next we show that the given minimization problem has a unique solution by contradiction. If the solution is not unique, let y_0 and y_1 be two distinct solutions. Then for $0 < t < 1$ we set $y_t = ty_1 + (1-t)y_1$. Since Ω is convex, we have $y_t \in \Omega$.

Thus

$$\begin{aligned}
\|y_0 - x\|_2^2 = \|y_1 - x\|_2^2 \leq \|y_t - x\|_2^2 = &\|t\,(y_1 - x) + (1-t)\,(y_0 - x)\|_2^2 \\
=&t^2\,\|y_1 - x\|_2^2 + (1-t)^2\,\|y_0 - x\|_2^2 + 2t(1-t)\,\langle y_1 - x, y_0 - x\rangle \\
=&t\,\|y_1 - x\|_2^2 + (1-t)\,\|y_0 - x\|_2^2 - \left(t - t^2\right)\|y_1 - x\|_2^2 \\
&- \left(1 - t - (1-t)^2\right)\|y_0 - x\|_2^2 + 2t(1-t)\,\langle y_1 - x, y_0 - x\rangle \\
=&t\,\|y_1 - x\|_2^2 + (1-t)\,\|y_0 - x\|_2^2 \\
&- t(1-t)\left(\|y_1 - x\|_2^2 + \|y_0 - x\|_2^2 - 2\,\langle y_1 - x, y_0 - x\rangle\right) \\
=&\,\|y_0 - x\|_2^2 - t(1-t)\,\|y_1 - y_0\|_2^2
\end{aligned}$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product on $\mathbb{R}^n$. Therefore $t(1-t)\,\|y_1 - y_0\|_2^2 \leq 0$ for $0 < t < 1$ and thus $\|y_1 - y_0\|_2^2 \leq 0$. So $y_1 = y_0$ which is a contradiction.

Overall, the minimization problem defined above has a unique minimizer. □

There are many different methods to solve the unconstrained optimization problem since generally, we treat this kind of problem as basic one. There are two most classical methods, Newton method [Newton and Colson, 1736] and the Method of Steepest Descent [Debye, 1909]. The former one, Newton method starts from an initial guess x_0 and defines a sequence $\{x_k\}$ iteratively according to some rule. It uses the tangent line of the objective function f at x_k to replace f and uses the root of $L(x) = 0$, where $L(x)$ is the updated $f(x)$ as the next iterate x_{k+1}. Finally, the iteration is terminted as long as the difference between x_k and x_{k+1} less than a preassigned small number. The later one, steepest descent is a basic gradient method, which decreases the value of the objective function in a direction of most rapid change. The change rate of a function f at x in the direction u, a unit vector in $\mathbb{R}$ is determined by the directional derivative. Therefore, at x the value of f decrease fastest in the direction $u = -\nabla f(x)/\|\nabla f(x)\|$, which leads to the gradient method: we update the x through the direction with the step length.

2.2.2 Equality Constrained Optimization

Constrained problems are usually more complicated since the solution is restricted on a boundary or in a feasible region. For equality constraints, the basic case is the linear equality constraints $A\boldsymbol{y} = \boldsymbol{b}$. Again, we consider an objective function $f : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. Let $A \in \mathbb{R}^{p\times m}$ and $b \in \mathbb{R}^p$. A is a set of p linear equations as constraints $A\boldsymbol{y} = \boldsymbol{b}$. The problem is defined as follows:

$$\begin{aligned}
y(x) \in \arg\min\nolimits_{y\in\mathbb{R}^m} \quad & f(x, y) \\
\text{subject to} \quad & A\boldsymbol{y} = \boldsymbol{b}
\end{aligned}$$

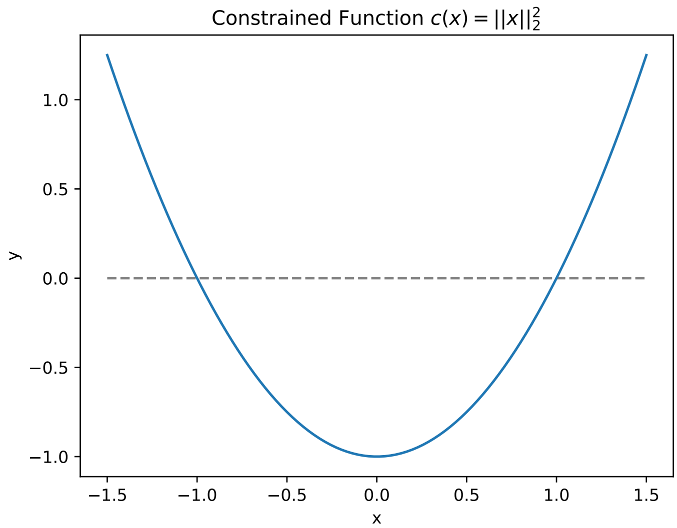


Figure 2.4: Constrained function $c(x) = \|x\|_2^2 - 1$

The derivative of $y(x)$ with respect to x is

$$\frac{dy(x)}{dx} = \left(H^{-1}A^T\left(AH^{-1}A^T\right)^{-1}AH^{-1} - H^{-1}\right)B \tag{2.4}$$

where $H = \partial^2 f(x,y)/\partial y(x)^2$ and $B = \partial^2 f(x,y)/\partial x \partial y(x)$.

The solution in 2.4 can be proved through the Lagrange multipliers [Bertsekas, 2014] in A.2.2. More generally, constraints can be non-linear. That means we cannot use A as a weight matrix for constrained parameters anymore. Therefore, we define the equality constraints problem using a set of m constraints functions $c(x,y)$:

$$\begin{aligned} y(x) \in \arg\min_{y \in \mathbb{R}^m} \quad & f(x,y) \\ \text{subject to} \quad & c_i(x,y) = 0, \quad i = 1, \ldots, m \end{aligned}$$

Solution for general multiple non-linear equality constraints is discussed in the chapter of deep declarative network nodes. Here, we are giving a simple example of non-linear equality constrained optimization problem.

For any given nonzero vector $y \in \mathbb{R}^n$, we define the minimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & -x^T y \\ \text{subject to} \quad & \|x\|_2^2 = 1 \end{aligned}$$

From the constraint defined above, we can write the constraint function as $c(x) = \|x\|_2^2 - 1$ and illustrate it in Fig 2.4. Differentiating $c(x)$ with respect to x, we get $\nabla c(x) = x \neq 0$. Therefore, it follows the definition of LFD 2.11 and the theorem of KKT 2.12, which means that every local minimizer of this problem is a KKT point.

Now we can write the Lagrangian function:

$$\mathcal{L}(x, \lambda) = -x^T y + \lambda \left(\|x\|_2^2 - 1 \right)$$

and the KKT conditions are:

$$\nabla_x \mathcal{L} = -y + 2\lambda x = 0, \quad \|x\|_2^2 = 1$$

From $-y + 2\lambda x = 0$ and $y \neq 0$ defined in the question, we must have $\lambda \neq 0$ and $x = y/2\lambda$. Combined with $\|x\|_2^2 = 1$, we get

$$4\lambda^2 = \|y\|_2^2 \Leftrightarrow \lambda = \pm \frac{\|y\|_2}{2}$$

Therefore, consequently, we have $x = \pm \frac{y}{\|y\|_2}$. For each x, we can compute its corresponding value of the objective function:

$$x = \frac{y}{\|y\|_2}, -x^T y = -\|y\|_2$$

$$x = -\frac{y}{\|y\|_2}, -x^T y = \|y\|_2$$

Obviously, the minimum is achieved $-\|y\|_2$ at $x = \frac{y}{\|y\|_2}$.

Algorithms for solving constrained problems are various. For basic linear programming, which means that all functions involved are linear, we can transform it into standard form with matrix A, then solve the problem using Lagrangian function based on the KKT condition.

Penalty method[Yeniay, 2005], a function determining when a point x is feasible or not, is used to replace the constrained problem with an unconstrained one. For a minimization problem $f(x)$, the penalty function $P(x)$ associated with a penalty parameter are introduced to combine with $f(x)$ and now we are going to solve a series of unconstrained problems. These problems have converged solutions of the original constrained problem.

2.2.3 Inequality Constrained Optimization

Similar to equality constrained problems, inequality constrained problem usually defined the solution in a feasible set. In general, the standard form of inequality constrained problem is negative constraints:

$$\begin{array}{rl} y(x) \in \arg\min_{y \in \mathbb{R}^m} & f(x, y) \\ \text{subject to} & c_i(x, y) \leq 0, \quad i = 1, \ldots, m \end{array}$$

Solutions for inequality constrained problems are various. According to the properties of inequality constraints, active and inactive constraints have different criteria. We aim to find the gradient of the optimal solution, $f\prime(x)$, based on the inequality constrained argmin function. Panier et al. [1988] proposed a globally convergent algorithm for solving the minimization of smooth objective function based on smooth inequality constraints. This algorithm is based on the Quasi-Newton [Dennis and Moré, 1977] iteration for the solution of the first order condition of the optimality in KKT. An updated verision of this algorithm, a new QP-free method demonstrated by Qi and Qi [2000], emphasizes the feasibility of all iterates. It reformulates the KKT optimality condition Fischer–Burmeister function [Jiang, 1999] for nonlinear complementarity problems. The classical solution is still based on the Lagrange multipliers. Bertsekas [2014] proposed that for one-sided inequality constrained problems, it cannot be converted to equality constrained problem. Therefore, it introduced the method minimizing the augumented Lagrangian with respect to x for various value of the Lagrange parameters, which is presented by Powell [1969] and Hestenes [1969]:

$$\begin{aligned}\bar{L}_c(x,z,\lambda,\mu) =& f(x) + \lambda' h(x) + \frac{1}{2}c|h(x)|^2 \\ &+ \sum_{j=1}^{r}\left\{\mu_j\left[g_j(x) + z_j^2\right] + \frac{1}{2}c\left|g_j(x) + z_j^2\right|^2\right\}\end{aligned}$$

The minimization of this augumented Lagrangian can be found through computing the first order derivative with respect to z explicitly for each fixed x.

More recently, Gould et al. [2016] introduced a method approximating the gradient of the inequality constrained problem based on ideas from interior-point methods [Boyd et al., 2004]. It gives a demonstration of log-barrier function, which transforms the original constrained problem into a unconstrained minimization problem.

$$\phi(x,y) = \sum_{i=1}^{m} \log\left(-c_i(x,y)\right) \tag{2.5}$$

$$\text{minimize}_y \quad tf(x,y) - \sum_{i=1}^{m} \log\left(-c_i(x,y)\right) \tag{2.6}$$

Equation 2.5 is the log-barrier function, which takes the sum of the logrithm for all constraints. Then substracting it in the unconstrained minimization problem approximates the original inequality constrained problem. t in Equation 2.6 is a scaling factor for duality gap control if the solution set is convex.

Similar to the solution for unconstrained and equality constrained problem, minimizing Equation 2.6 is based on the gradient and hessian of the log-barrier function. Therefore, we can compute the approximation of the inequality constrained objective function.

2.3 Differentiable Neural Network

If a problem is differentiable, that means the solution of this problem can be back-propagated. In neural networks, back-propagation [Goodfellow et al., 2016] is widely used to train the feedforward neural network, especially for supervised learning.

Therefore, in deep neural networks, we can treat constrained optimization as an individual layer. Recently, there are several works on end-to-end differentiable convex optimization in the neural network, since this type of layer provides inductive bias for different problems, which is very practical.

OptNet [Amos and Kolter, 2017] is a very classical differentiable layer neural network. Each layer in the end-to-end deep neural network is intergraded into optimization problems, which can capture and encode complex dependencies and constraints between hidden variables. It specifically considers the quadratic programs, which are general convex optimization problems. Similarly, SATNet [Wang et al., 2019], a differentiable maximum satisfiability solver, is also intergraded into end-to-end deep learning systems. Besides, it combines the solver with the traditional convolutional network. Both OptNet and SATNet are applied to solving the Sudoku puzzles, which is a very basic constrained logical problem. To make it more general and efficient, Agrawal et al. [2019] demonstrate an approach based on disciplined convex programs, which is a subclass of the classical convex optimization problems. The affine map introduced in this paper represents the disciplined parametrized program.

A fashion application in the differentiable network is the Perspective-n-Points (PnP) solver. Chen et al. [2020] present BPnP based on PnP solver, performing geometric optimization in computer vision tasks. It back-propagates gradient through PnP accurately and effectively since there is a differentiable function in the optimizer block. Besides, for blind PnP problems in the 3D computer vision task, Campbell et al. [2020] propose an end-to-end network based on the differentiating optimization solutions, which is robust and outperforming.

Apart from the above, the differentiable neural network has many practical and powerful applications. Amos and Yarats [2019] introduce a differentiable variant cross-entropy method for non-convex optimization objective function. Again, due to the differentiable feature of the network, the output of the cross-entropy method is differentiable with respect to the parameters in the objective function, even it is non-convex. In 3D reconstruction tasks, some implicit shape and texture are difficult to represent. Hence, Niemeyer et al. [2020] introduce a differentiable rendering formulation, which makes the network learn them from input images directly since implicit differentiation can learn the depth gradients. Also, some research has been done to simplify the differentiable neural network since the computational cost and complexity of the differential operators can be very high in different tasks. The architecture proposed by Chen and Duvenaud [2019] is cheap and efficient, which sets the Jacobian matrix into diagonal and hollow. It also changes the backward progress into automatic differentiation, which is more effective and lightweight.

2.4 Summary

In Chapter 2, firstly, introduce the numerical optimization briefly with some necessary conditions and theorems. For the general convex optimization problem, the existence of the local or global optimizers can be determined by the feasible set. Next, the optimality of both unconstrained and constrained problems are discussed. For unconstrained problems, we only need to follow the necessary and sufficient conditions to find the global minimum of the solution. For constrained problems, it should also satisfy the KKT condition. Then we compare existing algorithms on the solution to these problems. Here, for constrained optimization problems, we have to consider the linearity of constraints. Specifically, the activity of inequality constraints can also be solved with different algorithms. Finally, since this thesis is based on the end-to-end differentiable network, some related works of the application are described since these works inspire the deep declarative network in the next chapter. In the next chapter, we are going to describe the deep declarative network in detail with examples.

Chapter 3

Deep Declarative Network

In this chapter, we will cover the structure and nodes in the deep declarative network: from its learning process to the back-propagation.

Before delving into the details of the back-propagation in different constraints cases, we give an overview of the deep declarative network in Section 3.1. In particular, the basic structure of the network and the details of declarative nodes are described according to Gould et al. [2019]. The learning progress of the network is also given. We hope this will give readers a better sense of what is the deep declarative network and how it works.

In Section 3.2, we present the details of the back-propagation in different constrained problems. The gradient computation results are based on the implicit differentiation and different in constrained problems. We discuss this part based on the regular solution and compare it with the general solution in the previous chapter.

Next we present the examples of constrained optimization problems with both linear and non-linear, equality and inequality constraints in Section 3.3. We also provide more implementation details of the deep declarative nodes.

Finally, we summarize the deep declarative network and its solution in different constrained problems under the regular point.

3.1 An Overview of Deep Declarative Network

3.1.1 Declarative Node

In deep declarative network, it defines the solution of a constrained optimization problem with parameter $x \in \mathbb{R}^n$ as the output of each node $y \in \mathbb{R}^m$. The general optimization problem can be defined as

$$y \in \underset{u \in C}{\arg\min} f(x, u) \tag{3.1}$$

where f is the objective function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, and $C \in \mathbb{R}^m$ is the set of constraints parameterized by x.

Apart from the traditional forward processing mapping node, deep declarative node does not explicitly define the transforming function from the input to the out-

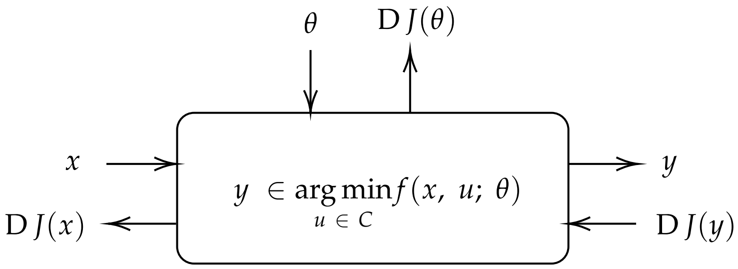


Figure 3.1: End-to-end learnable declarative node

put. It defines the input-output relationship implicitly by an objective and constriants optimization problem, where the solution of the problem is the output.

Figure 3.1 shows the forward and backward pass of the declarative node. In the forward evaluation pass, the output of the declarative y is computed as the solution of some minimization problem $f(x, u; \theta)$. We use D to denote the total derivative with respect to the independent variables. Therefore, in the backward pass, the gradient of the global objective function with respect to the output $\mathrm{D}\,J(y)$ is back-propagated. Its value is computed through the chain rule based on the gradients with respect to the input $\mathrm{D}\,J(x)$ and parameters $\mathrm{D}\,J(\theta)$.

Since the definition of deep declarative nodes is very general, it can be embedded within another network for solving subproblems such as robust fitting. However, we may not be able to find the gradient when the feasible set is discrete, or the declarative node is low efficiency to evaluate. As non-regular solution cases, the nonexistent gradient problem will be discussed in the next part. In the next subsection, the learning details of the deep declarative network are described.

3.1.2 Learning

Since in declarative nodes, there is no explicit forward function defined, we can directly compute the optimal solution y through some algorithms. Under this assumption, when we performing the back-propagation, we can compute the gradient of the output from each node with respect to the corresponding input through the implicit differentiation directly. This can be treated as a bi-level optimization problem[Bard, 1998] where the parameterized constraints as a lower-level problem blinds variables in the objective function, an upper-level problem. Combining the schematic illustration in Figure 3.1, the problem can be defined formally as

$$\begin{array}{ll} \text{minimize} & J(x, y) \\ \text{subject to} & y \in \arg\min_{u \in C} f(x, u) \end{array} \tag{3.2}$$

We may have additional layers to make the objective function $J(x, y)$ depend on y, which is a function of x. In general, it is the sum of loss terms and regularization terms. We can solve this minimization problem through the gradient descent as follows:

$$\mathrm{D}\, J(x,y) = \mathrm{D}_X\, J(x,y) + \mathrm{D}_Y\, J(x,y)\, \mathrm{D}\, y(x) \tag{3.3}$$

where $\mathrm{D}_X\, J(x,y)$ is the partial derivatives of $J(x,y)$ with respect to x and $\mathrm{D}_Y\, J(x,y)$ is the partial derivatives of $J(x,y)$ with respect to y. We used to use D_X and D_Y to denote the partial derivatives. We decompose the total derivatives of $J(x,y)$ as the sum of the partial derivatives with the chain rule. In application, we can consider it as the sum of gradients for losses on training examples.

The lower-level objective function f can be simpler. If it is the only term involving y in the upper-level objective function J, that means $J(x,y) = g(x, f(x,y))$ and the lower-level problem is actually unconstrained with $u \in C = \mathbb{R}^m$. Under this condition, the calculation of the gradient can be expanded using chain rule through both $\mathrm{D}_X\, J(x,y)$ and $\mathrm{D}_Y\, J(x,y)$:

$$\begin{aligned} \mathrm{D}\, J(x,y) &= \mathrm{D}_X\, g(x,f) + \mathrm{D}_F\, g(x,f)\, (\mathrm{D}\, f + \mathrm{D}_Y\, f\, \mathrm{D}\, y) \\ &= \mathrm{D}_X\, g(x,f) + \mathrm{D}_F\, g(x,f)\, \mathrm{D}\, f \end{aligned} \tag{3.4}$$

where $\mathrm{D}_Y\, f(x,y) = 0$ since y is the minimum of $f(x,y)$ and $f(x,y)$ is an unconstrained problem, its partial derivative should be zero.

3.2 Back-propagation Through Declarative Nodes

Let us focus back on the more general case with y involving in different terms. The backward pass is different in different sub-classes of declarative nodes. We consider three common cases based on Equation 3.1.

3.2.1 Unconstrained

Firstly, the most basic case is the unconstrained problem. Consider a function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, we have

$$y \in \underset{u \in C}{\arg\min} f(x,u) \tag{3.5}$$

We make the assumption that the solution of this problem, $y(x)$ exists, and in the neighborhood of the point $(x, y(x))$, f is second-order differentiable. Therefore, we can compute the derivative of y with respect to x is

$$\mathrm{D}\, y(x) = -H^{-1}B$$

where $H = \mathrm{D}^2_{YY}\, f(x, y(x)) \in \mathbb{R}^{m \times m}$ is the second-order derivative of f with respect to y, and it is a non-singular matrix. $B = \mathrm{D}^2_{XY}\, f(x, y(x)) \in \mathbb{R}^{m \times n}$ is the second-order derivative of f with respect to y and x (the derivative of $D_Y f(x,y)$ with respect to x).

The proof of this solution is similar to the proof of Equation 2.3, setting the

3.2.2 Equality Constrained

3.2.3 Inequatlity Constrained

3.3 Examples of Declarative Nodes

3.3.1 Implementation Details

3.3.2 Equality Constrained

3.3.3 Inequatlity Constrained

3.4 Summary

Chapter 4

The Future of Declarative Nodes

Same as the last chapter, introduce the motivation and the high-level picture to readers, and introduce the sections in this chapter.

4.1 Improvements of the Optimization

4.2 Applications in Computer Vision Tasks

Part II

Deep Declarative Nodes: Non-regular Solution

Chapter 5

An Overview of Regular and Non-regular Solution

5.1 Problems in Regular Deep Declarative Nodes

5.2 Related Work in Non-regular Solution

5.2.1 Overdetermined System

5.2.2 Rand Deficiency

5.2.3 Non-convex Problems

Table 5.1 shows how to include tables and Figure 5.1 shows how to include codes.

Architecture	Pentium 4	Atom D510	i7-2600
Model	P4D 820	Atom D510	Core i7-2600
Technology	90nm	45nm	32nm
Clock	2.8GHz	1.66GHz	3.4GHz
Cores $\times$ SMT	2×2	2×2	4×2
L2 Cache	1MB $\times$ 2	512KB $\times$ 2	256KB $\times$ 4
L3 Cache	none	none	8MB
Memory	1GB DDR2-400	2GB DDR2-800	4GB DDR3-1066

Table 5.1: Processors used in our evaluation.

```
int main(void)
{
  printf("Hello␣World\n");
  return 0;
}
```

(a)

```
void main(String[] args)
{
  System.out.println("Hello␣World");
}
```

(b)

Figure 5.1: Hello world in Java and C.

Chapter 6

Solutions of Non-regular Point

6.1 Overdetermined System

6.1.1 Least-Squared Method

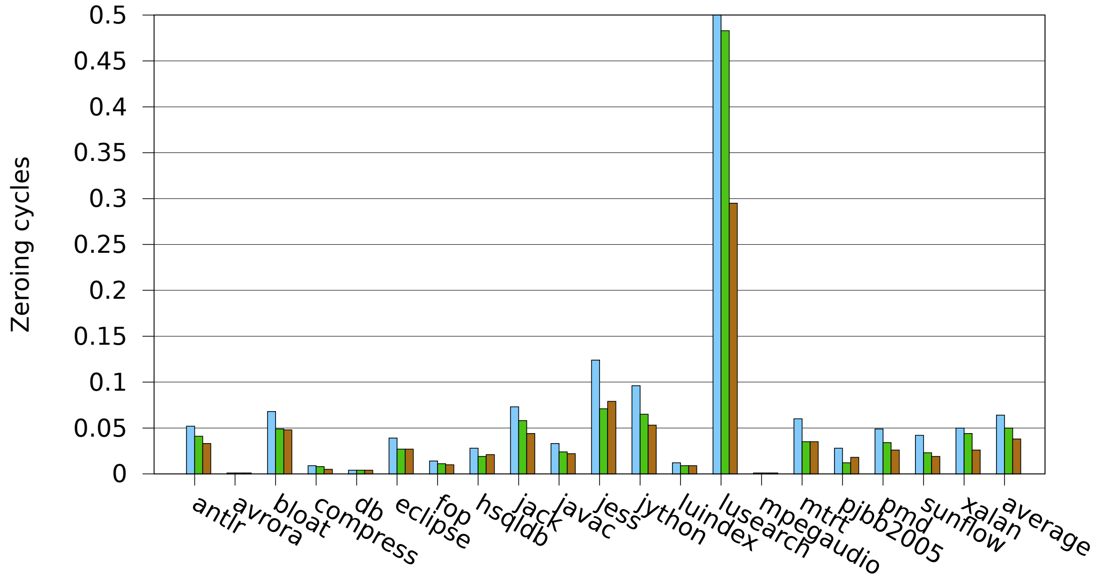
6.2 Conjugate Gradient and Preconditioning

6.3 Rank Deficiency

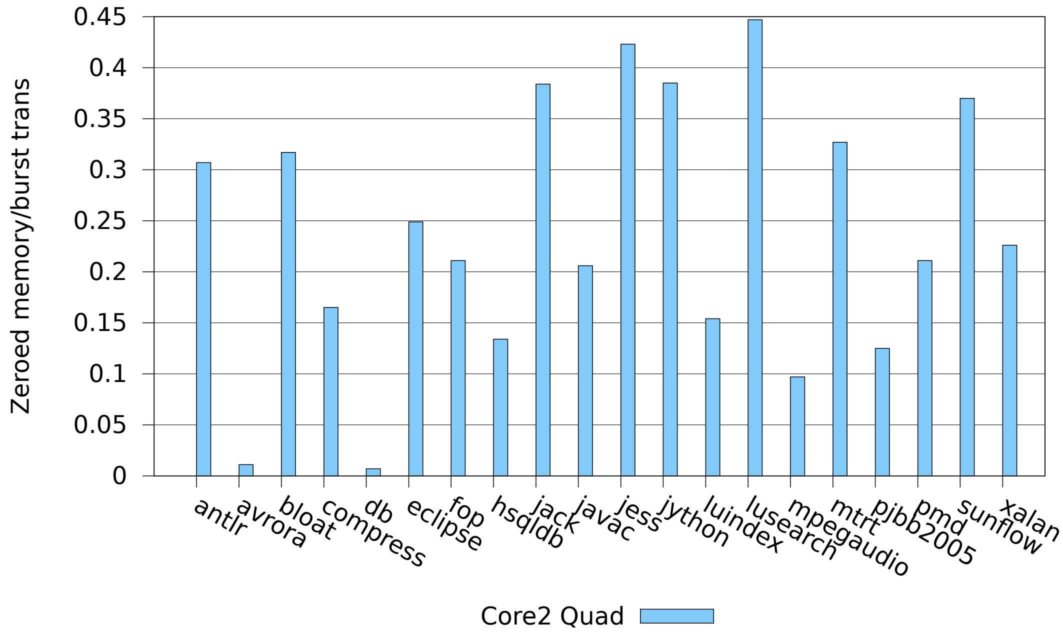
6.4 Non-convex Problems

Here is the example to show how to include a figure. Figure 6.1 includes two subfigures (Figure 6.1(a), and Figure 6.1(b));

6.5 Summary



(a) Fraction of cycles spent on zeroing



(b) BytesZeroed / BytesBurstTransactionsTransferred

Figure 6.1: The cost of zero initialization

Chapter 7

Conclusion

Summary your thesis and discuss what you are going to do in the future in Section 7.1.

7.1 Future Work

Good luck.

Bibliography

Agrawal, A.; Amos, B.; Barratt, S.; Boyd, S.; Diamond, S.; and Kolter, J. Z., 2019. Differentiable convex optimization layers. In *Advances in neural information processing systems*, 9562–9574. (cited on page 18)

Amos, B. and Kolter, J. Z., 2017. Optnet: Differentiable optimization as a layer in neural networks. *arXiv preprint arXiv:1703.00443*, (2017). (cited on pages 12 and 18)

Amos, B. and Yarats, D., 2019. The differentiable cross-entropy method. *arXiv preprint arXiv:1909.12830*, (2019). (cited on page 18)

Bard, J. F., 1998. *Practical Bilevel Optimization: Algorithms and Applications*, vol. 30. Springer Science & Business Media. (cited on page 22)

Bertsekas, D. P., 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press. (cited on pages 12, 15, and 17)

Boyd, S.; Boyd, S. P.; and Vandenberghe, L., 2004. *Convex optimization*. Cambridge university press. (cited on page 17)

Campbell, D.; Liu, L.; and Gould, S., 2020. Solving the blind perspective-n-point problem end-to-end with robust differentiable geometric optimization. *arXiv preprint arXiv:2007.14628*, (2020). (cited on page 18)

Chen, B.; Parra, A.; Cao, J.; Li, N.; and Chin, T.-J., 2020. End-to-end learnable geometric vision by backpropagating pnp optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8100–8109. (cited on page 18)

Chen, R. T. and Duvenaud, D. K., 2019. Neural networks with cheap differential operators. In *Advances in Neural Information Processing Systems*, 9961–9971. (cited on page 18)

Debye, P., 1909. Näherungsformeln für die zylinderfunktionen für große werte des arguments und unbeschränkt veränderliche werte des index. *Mathematische Annalen*, 67, 4 (1909), 535–558. (cited on page 14)

Dennis, J. E., Jr and Moré, J. J., 1977. Quasi-newton methods, motivation and theory. *SIAM review*, 19, 1 (1977), 46–89. (cited on page 17)

Goodfellow, I.; Bengio, Y.; Courville, A.; and Bengio, Y., 2016. *Deep learning*, vol. 1. MIT press Cambridge. (cited on page 17)

Gould, S.; Fernando, B.; Cherian, A.; Anderson, P.; Cruz, R. S.; and Guo, E., 2016. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, (2016). (cited on pages 12 and 17)

Gould, S.; Hartley, R.; and Campbell, D., 2019. Deep declarative networks: A new hope. *arXiv preprint arXiv:1909.04866*, (2019). (cited on pages xii, 1, 21, and 40)

Hestenes, M. R., 1969. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4, 5 (1969), 303–320. (cited on page 17)

Jiang, H., 1999. Global convergence analysis of the generalized newton and gauss-newton methods of the fischer-burmeister equation for the complementarity problem. *Mathematics of Operations Research*, 24, 3 (1999), 529–543. (cited on page 17)

Newton, I. and Colson, J., 1736. *The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines... Translated from the Author's Latin Original Not Yet Made Publick. To which is Subjoin'd a Perpetual Comment Upon the Whole Work... by J. Colson*. (cited on page 14)

Niemeyer, M.; Mescheder, L.; Oechsle, M.; and Geiger, A., 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3504–3515. (cited on page 18)

Nocedal, J. and Wright, S., 2006. *Numerical optimization*. Springer Science & Business Media. (cited on pages 6 and 7)

Oman, G., 2017. A short proof of the bolzano-weierstrass theorem. *The College Mathematics Journal*, (2017). (cited on page 6)

Panier, E. R.; Tits, A. L.; and Herskovits, J. N., 1988. A qp-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Control and Optimization*, 26, 4 (1988), 788–811. (cited on page 17)

Powell, M. J., 1969. A method for nonlinear constraints in minimization problems. *Optimization*, (1969), 283–298. (cited on page 17)

Qi, H.-D. and Qi, L., 2000. A new qp-free, globally convergent, locally superlinearly convergent algorithm for inequality constrained optimization. *SIAM Journal on Optimization*, 11, 1 (2000), 113–132. (cited on page 17)

Wang, P.-W.; Donti, P. L.; Wilder, B.; and Kolter, Z., 2019. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. *arXiv preprint arXiv:1905.12149*, (2019). (cited on page 18)

Yeniay, Ö., 2005. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and computational Applications*, 10, 1 (2005), 45–56. (cited on page 16)

Appendix

A An Overview of Numerical Optimization

A.1 Theory of Optimization

A.1.1 Proof of Theorem 2.3

Proof. Let
$$m := \inf\{f(x) : x \in \Omega\}$$

By the definition of m we may pick a sequence $\{x_k\} \subset \Omega$ with $f(x_k) \to m$ as $k \to \infty$. Because Ω is compact, we can extract a convergent subsequence $\left\{x_{k_j}\right\}$ from $\{x_k\}$. Let $x^* \in \Omega$ denote the limit point of $\left\{x_{k_j}\right\}$. Since f is continuous, $f(x^*) = \lim_{j\to\infty} f\left(x_{k_j}\right) = m$. Thus m is finite and x^* is a global minimizer of f on Ω.

When $\Omega = \mathbb{R}^n$, we need to impose conditions on f at infinity to guarantee the existence of a global minimizer. □

A.1.2 Proof of Theorem 2.4

Proof. Let $m := \inf\{f(x) : x \in \mathbb{R}^n\}$, and take a sequence $\{x_k\}$ such that
$$f(x_k) \to m \quad \text{as } k \to \infty.$$

Since f is coervice, $\{x_k\}$ must be bounded; otherwise it has a subsequence $\left\{x_{k_j}\right\}$ with $\left\|x_{k_j}\right\| \to \infty$ as $j \to \infty$, and hence $m = \lim_{j\to\infty} f\left(x_{k_j}\right) = +\infty$, a contradition.

Thus there is $r > 0$ such that
$$\{x_k\} \subset \{x \in \mathbb{R}^n : \|x_k\| \le r\}.$$

Because $\{x \in \mathbb{R}^n : \|x\| \le r\}$ is compact, $\{x_k\}$ has a convergent subsequence $\left\{x_{k_j}\right\}$

with $x_{k_j} \to x^*$ as $j \to \infty$. In view of the continuity of f, we have

$$f(x^*) = \lim_{j\to\infty} f\left(x_{k_j}\right) = m$$

Therefore m is finite and f achieves its minimum on $\mathbb{R}^n$ at x^* $\square$

A.1.3 Proof of Theorem 2.6

Proof. We may assume that $\alpha > f_* := \inf\{f(x) : x \in \mathbb{R}^n\}$. Let $\{x_k\}$ be a minimizing sequence for f, i.e.

$$f(x_k) \to f_* \quad \text{as } k \to \infty$$

Then there is an N such that $f(x_k) \le \alpha$ for all $k \ge N$, that is, $x_k \in D$ for all $k \ge N$. Since D is compact, $\{x_k\}_{k=N}^{\infty}$ has a convergent subsequence $\left\{x_{k_j}\right\}$ with $x_{k_j} \to x_* \in D$ as $j \to \infty$. In view of the lower semi-continuity of f, we have

$$f(x_*) \le \lim_{j\to\infty} f\left(x_{k_j}\right) = f_*$$

By the definition of f_* we must have $f(x_*) = f_*$. Therefore f achieves its minimum on $\mathbb{R}$ at x_*. $\square$

A.1.4 Proof of Theorem 2.8

Proof. (NC1): First, recall that for any $v \in \mathbb{R}^n$ there holds

$$v^T \nabla f(x^*) = D_v f(x^*) = \lim_{t\searrow 0} \frac{f(x^* + tv) - f(x^*)}{t}.$$

Since x^* is a local minimizer, we have

$$f(x^* + tv) - f(x^*) \ge 0 \quad \text{for small } |t|.$$

Therefore

$$v^T \nabla f(x^*) \ge 0 \quad \text{for all } v \in \mathbb{R}^n.$$

In particular this implies $(-v)^T \nabla f(x^*) \ge 0$ and thus

$$v^T \nabla f(x^*) \le 0 \quad \text{for all } v \in \mathbb{R}^n.$$

Therefore $v^T \nabla f(x^*) = 0$ for all $v \in \mathbb{R}^n$. Taking $v = \nabla f(x^*)$ gives $\|\nabla f(x^*)\|^2 = 0$ which shows that $\nabla f(x^*) = 0$ $\square$

Proof. (NC2): Recall that for any $v \in \mathbb{R}^n$ and small $t > 0$ there is $0 < s < 1$ such that

$$f(x^* + tv) = f(x^*) + tv^T \nabla f(x^*) + \frac{1}{2} t^2 v^T \nabla^2 f(x^* + stv)\, v.$$

Since x^* is a local minimizer of f, we have $f(x^* + tv) \geq f(x^*)$ and $\nabla f(x^*) = 0$ by (NC1). Therefore

$$\frac{1}{2}t^2 v^T \nabla^2 f(x^* + stv)\, v = f(x^* + tv) - f(x^*) \geq 0.$$

This implies that

$$v^T \nabla^2 f(x^* + stv)\, v \geq 0.$$

Taking $t \to 0$ gives

$$v^T \nabla^2 f(x^*)\, v \geq 0 \quad \text{for all } v \in \mathbb{R}^n$$

i.e. $\nabla^2 f(x^*)$ is semi-definite. □

Proof. (SC1): Since $\nabla^2 f(x$ is continuous and $\nabla^2 f(x^*) \geq 0$, we can find $r > 0$ such that

$$B_r(x^*) \subset \Omega \quad \text{and} \quad \nabla^2 f(x) > 0 \text{ for all } x \in B_r(x^*).$$

By Taylor's formula we have

$$f(x) = f(x^*) + \nabla f(x^*) \cdot (x - x^*) + \frac{1}{2}(x - x^*)^T \nabla^2 f(\hat{x})(x - x^*)$$

where $\hat{x} := x^* + t(x - x^*)$ for some $0 < t < 1$.

It is clear that $\hat{x} \in B_r(x^*)$ and hence $\nabla^2 f(\hat{x}) > 0$ which implies that

$$(x - x^*)^T \nabla^2 f(\hat{x})(x - x^*) > 0 \quad \text{for } x \neq x^*$$

Consequently

$$f(x) > f(x^*) + \nabla f(x^*) \cdot (x - x^*)$$

for all $x \in B_r(x^*)$ with $x \neq x^*$.

Since $\nabla f(x^*) = 0$, we can obtain $f(x) > f(x^*)$ for all $x \in B_r(x^*)$ with $x \neq x^*$. □

A.1.5 Proof of Lemma 2.10

Proof. For $d \in T_{x^*}\mathscr{F}$, we have $z_k \subset \mathscr{F}$ and t_k such that

$$z_k \to x^*, \quad 0 < t_k \to 0 \quad \text{and} \quad \frac{z_k - x^*}{t_k} \to d$$

as $k \to \infty$. As $f(x^*) \leq f(z_k)$, by Taylor's formula we have

$$\begin{aligned} f(x^*) &\leq f(z_k) = f(x^* + (z_k - x^*)) \\ &= f(x^*) + (z_k - x^*)^T \nabla f(x^*) + \frac{1}{2}(z_k - x^*)^T \nabla^2 f(\hat{z}_k)(z_k - x^*) \end{aligned}$$

where $\hat{z}_k$ is a point on the line segment joining x^* and z_k. This implies that

$$0 \leq \left(\frac{z_k - x^*}{t_k}\right)^T \nabla f(x^*) + \frac{1}{2}(z_k - x^*)^T \nabla^2 f(\hat{z}_k)\left(\frac{z_k - x^*}{t_k}\right)$$

Letting $k \to \infty$ gives $d^T \nabla f(x^*) \geq 0$ □

A.2 Solution of Unconstrained and Constrained Optimization Problems

A.2.1 Proof of Equation 2.3

Proof. Firstly, for any optimal y, according to the first-order optimality condition, we have

$$\frac{df(x,y)}{dy} = \mathbf{0} \in \mathbb{R}^{1\times m}$$

Then from the implicit function theorem, rearranging and differentiating both sides we have

$$\begin{aligned} \mathrm{D}(\frac{df(x,y)}{dy})^T &= \mathbf{0} \in \mathbb{R}^{m\times n} \\ &= \frac{\partial^2}{\partial x \partial y} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y)\frac{dy(x)}{dx} \\ \frac{dy(x)}{dx} &= -[(\frac{\partial^2}{\partial y^2})f(x,y)]^{-1}(\frac{\partial^2}{\partial x \partial y})f(x,y) \end{aligned}$$

□

A.2.2 Proof of Equation 2.4 [Gould et al., 2019]

Proof. According to the definition of Lagrange multipliers, we can define the Lagrangian:

$$\mathcal{L}(x,y,\lambda) = f(x,y) - \sum_{i=1}^{p} \lambda_i (A_i y_i - b_i)$$

We are going to find the stationary point (y, λ) for this lagrangian. Therefore, we calculate the derivative of $\mathcal{L}$ with respect to y and λ separately:

$$\frac{\partial}{\partial y} f(x,y) - \sum_{i=1}^{p} \lambda_i \frac{\partial}{\partial y}(A_i y_i - b_i) = 0 \tag{1}$$

$$A\boldsymbol{y} - \boldsymbol{b} = 0 \tag{2}$$

Since y is the optimal point, we have $\frac{\partial}{\partial y} f(x,y) = 0$, which can be an unconstrained problem or it is orthogonal to the constraint surface. For unconstrained cases, we can set $\lambda = 0$ directly. For the orthogonal case, from Equation 1, we have

$$\frac{\partial}{\partial y} f(x,y) = \sum_{i=1}^{p} \lambda_i \frac{\partial}{\partial y}(A_i y_i - b_i) = \lambda^T A$$

Now we are going to calculate the derivative of the Lagrangian with respect to x for both 1 and 2:

$$\frac{\partial^2}{\partial x \partial y} f(x,y) + \frac{\partial^2}{\partial y^2} f(x,y) \mathrm{D}y - \frac{\partial}{\partial y}(A\boldsymbol{y} - \boldsymbol{b})^T \mathrm{D}\lambda = 0 \tag{3}$$

$$\frac{\partial}{\partial x}(A\boldsymbol{y} - \boldsymbol{b}) + \frac{\partial}{\partial y}(A\boldsymbol{y} - \boldsymbol{b}) \mathrm{D}y = 0 \tag{4}$$

Solving 3 and 4, we get:

$$\mathrm{D}y(x) = \left(H^{-1} A^T \left(A H^{-1} A^T \right)^{-1} A H^{-1} - H^{-1} \right) B$$

where

$$H = \frac{\partial^2}{\partial y^2} f(x,y), \quad B = \frac{\partial^2}{\partial x \partial y} f(x,y)$$

$\square$