Comparing Perturbed Optimizer and Barrier Function Method for Differentiating Linear Programs

Mingrui Zhao
Supervised by: Prof. Stephen Gould

28 Oct 2021

Table of Contents

Table of Contents

Deep Learning

- Applied in image classification, voice recognition and so much more.
- Core of deep learning: Artificial Neural Networks.
  - Directed graph of parametrised processing nodes [1]

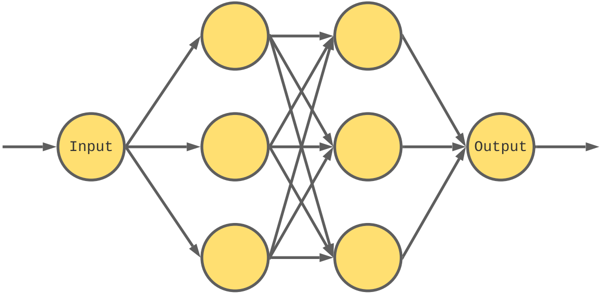


Figure: A conceptual sketch of a neural network

- Learning: Tune node parameters
- Done through back propagation
- Require node gradient

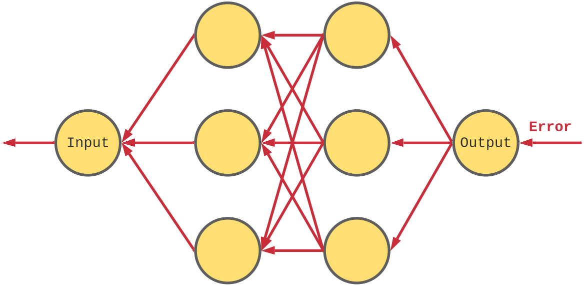


Figure: Conceptual sketch of back propagation of a neural network

Declarative Nodes

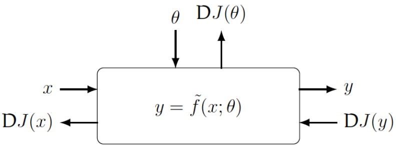


Figure: Imperative Node [1]

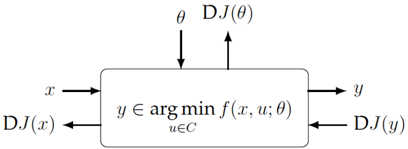


Figure: Declarative Node [1]

- All imperative nodes can be expressed as declarative nodes
- Declarative node has functions beyond the capability of imperative nodes, such as hard-code constraints into nodes.

Linear Program (LP)

- Declarative node fails on Linear Program

Definition

Optimisation problem with linear objective function and constraints. The canonical form of a Linear Program is

$$\text{minimize } c^T x$$
$$\text{subject to } Ax \preceq b$$

where $c, x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$

- The solution to a Linear Program always lies on a vertex of its feasible region

Toy Example

$$\begin{aligned} \text{minimize} \quad & \theta x_1 - 4x_2 \\ \text{subject to} \quad & x_1 + x_2 \leq 30 \\ & x_2 \geq 0, x_1 \geq 1 \end{aligned} \tag{1}$$

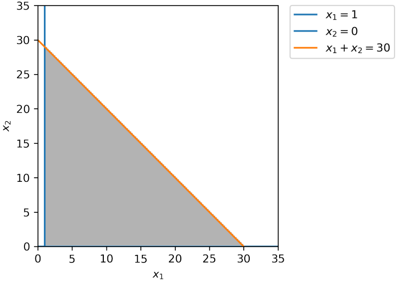


Figure: Feasible region

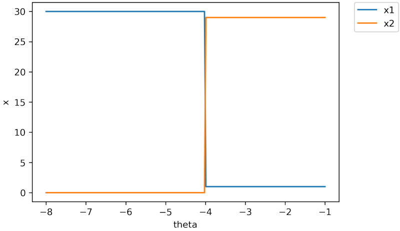


Figure: Solution of LP versus θ

A constant zero gradient defects back propagation, hence model training

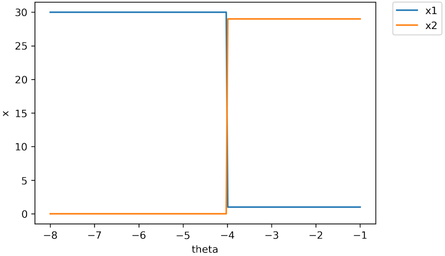


Figure: Solution of LP versus θ

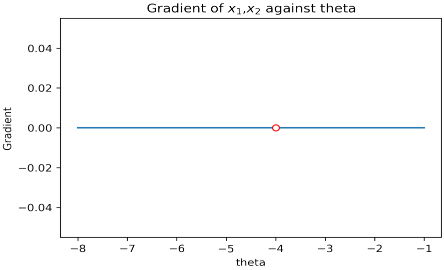


Figure: Gradient of solution versus θ

Research Question

Barrier function method v.s. Perturbed optimizer method

Table of Contents

Barrier Function Method

Combine constraints functions with the objective function through log barrier function.

Definition

Consider the optimisation problem

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \preceq b \end{aligned}$$

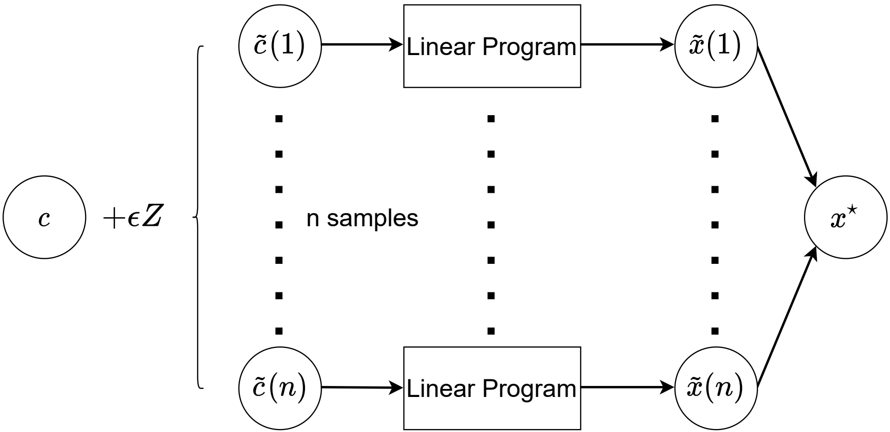
Applying barrier function method transforms this LP to an unconstrained optimisation problem

$$\text{minimize} \quad tc^T x - \sum_{i=1}^{m} \log\left(b_i - a_i^T x\right)$$

where a_i^T is the i_{th} row of A

$\frac{\partial x}{\partial c} = -D_{xx}^{-1} f(x, c) D_{xc} f(x, c)$ [1]

Perturbed Optimiser Method



$\frac{\partial x}{\partial c} = \mathbf{E}\left[x^*(c+\epsilon Z)\nabla_z \nu(Z)^\top/\epsilon\right]$ where Z is the noise, $\nu(z)$ is a function defined from noise distribution and ϵ is the temperature parameter of noise [2].

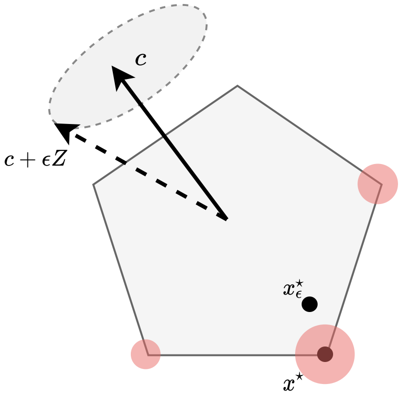


Figure: Geometric interpretation of perturbed optimizer

Table of Contents

Differentiating Toy LP

Recall the toy LP problem shown earlier

Toy Example

$$\begin{aligned} \text{minimize} \quad & \theta u_1 - 4u_2 \\ \text{subject to} \quad & u_1 + u_2 \leq 30 \\ & u_2 \geq 0, u_1 \geq 1 \end{aligned} \tag{2}$$

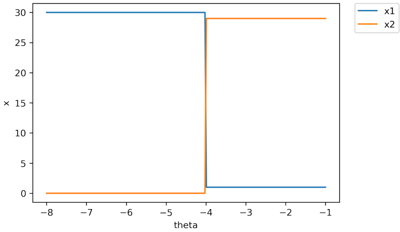


Figure: Solution of LP versus θ

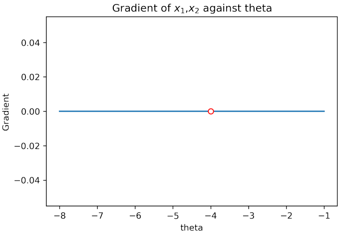


Figure: Gradient of solution versus θ

t = 0.5
Avg. time cost for sol/grad = 0.01s/0.0001s

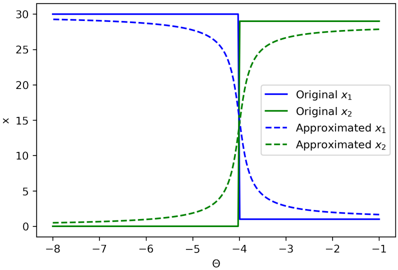


Figure: u against θ

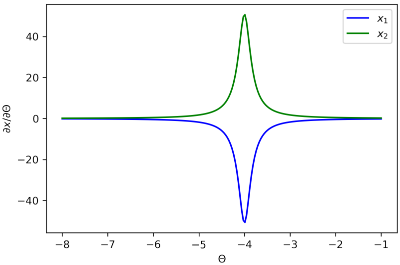


Figure: Gradient of u against θ

Differentiating Toy LP – Perturbed Optimizer Method

$\sigma = 0.7$, Gumbel Noise, Sample Size $= 2{,}000$
Avg. time cost for sol/grad $= 14.82$s$/0.0002$s

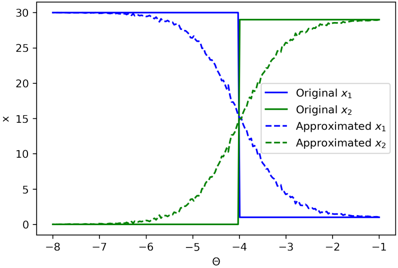


Figure: u against θ

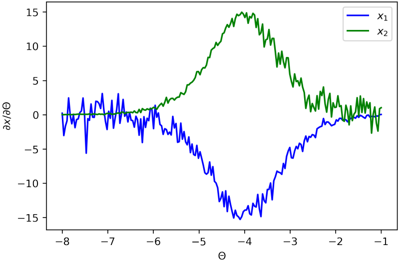


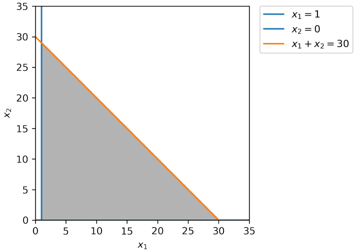
Figure: Gradient of u against θ

Back Propagation through LP Declarative Node

LP Declarative Node

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & u_1 + u_2 \leq 30 \\ & u_2 \geq 0, u_1 \geq 1 \end{aligned} \tag{3}$$

Initialise $c = [-8, -4]$ with $x^\star = [30, 0]$
Aim to update c with approximated gradient to achieve $x^\star = [1, 29]$



Back Propagation through LP Declarative Node
Barrier Function Method

Loss $= \|\bar{x} - [1, 29]\|_2^2$, learning rate $= 0.01$, t=0.1

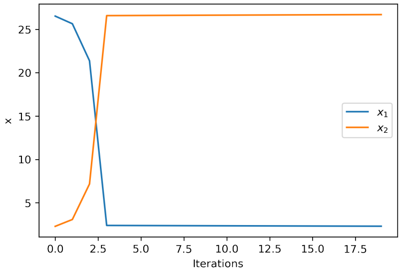


Figure: $\bar{x}$ over training iterations

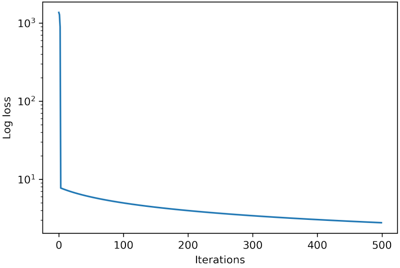


Figure: Log loss of declarative node

Back Propagation through LP Declarative Node
Perturbed Optimizer Method

Loss $= \|\bar{x} - [1, 29]\|_2^2$, learning rate $= 0.01$
$\epsilon = 0.5$, Sample Size $= 200$, Noise type $=$ Gumbel

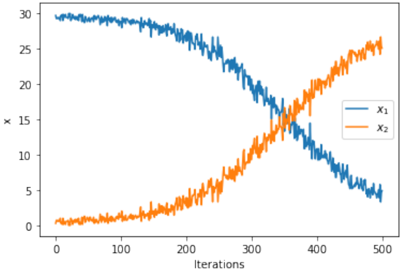


Figure: $\bar{x}$ over training iterations

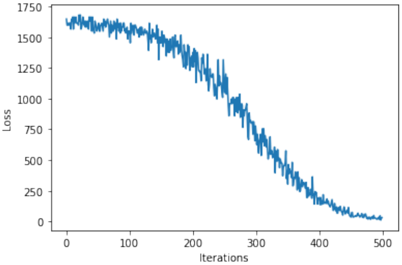


Figure: Loss of declarative node

Trainability of LP declarative node on CIFAR10

- CIFAR10: An image dataset contains 10 classes of object
- Initialise a Convolutional Neural Network and embed an LP declarative node.
- Perform Image classification tasks on CIFAR10
- Not aiming for state of the art results, but to demonstrate the ability of incorporating differentiable LP into an end-to-end learnable model.

Trainability of LP declarative node on CIFAR10

	Original CNN	Plain LP
Mean Average Accuracy	63%	10%

	Perturbed Optimizer	Barrier Function Method
Mean Average Accuracy	30%	N/A

- Proves the trainability of LP declarative node within a complete network
- Barrier function method failed due to limited computational resource
  - Determine the gradient of barrier function method requires inverting a Hessian matrix, in this model the matrix is 400×400 in dimension.

Table of Contents

Conclusion

Barrier Function Method

- Low computational cost for low dimensional LP
- Exponentially raising computational cost for high dimensional LP
- Deterministic approximation
- Smooth gradient approximation
- Require optimisation problem solver capable for log barrier terms

Perturbed Optimiser Method

- Computational cost is sample size dependent, generally it is high
- Relatively Lower computational cost for high dimensional LP
- Stochastic approximation
- Non-smooth gradient approximation, can be improved by increasing sample size
- Capable with any solver

Future Works

- Determine theoretical equivalency between perturbed optimizer and barrier function methods
- Find constructive LP declarative nodes for model training, aim for state of the art results

Bibliorgraphy

- S. Gould, R. Hartley, and D. Campbell, "Deep declarative networks: A new hope," *arXiv preprint arXiv:1909.04866*, 2019.
- Q. Berthet, M. Blondel, O. Teboul, M. Cuturi, J.-P. Vert, and F. Bach, "Learning with differentiable perturbed optimizers," *arXiv preprint arXiv:2002.08676*, 2020.
- Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- J. Abernethy, C. Lee, and A. Tewari, "Perturbation techniques in online learning and optimization," *Perturbations, Optimization, and Statistics*, p. 233, 2016.
- A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research),"