Report of Project 1
A Review of Convexified Convolutional Neural Network

Xiaodong Jia

June 6, 2017

Contents

1 Introduction

In this report, we analysis the convexification of a two layer neural network based on this paper[1], make a review of the related works and use some new methods to solve the same problem.

2 A Brief Summary of the Main Ideas

A convolutional neural network(CNN) can be written as a function $f(x)$,

$$f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_2},$$

where d_0 is the dimension of input vectors x, d_2 is the number of classes.
Particularly, for a two layer convolutional neural network, the following form separates the trainable parameters and other terms,

$$f^A(x) := (\mathrm{tr}(Z(x)A_1), ..., \mathrm{tr}(Z(x)A_{d_2})),$$

where A denotes all trainable parameters, and Z only depends on the inputs.
If the loss function $\mathcal{L}(f; y)$ is convex about f, $\mathcal{L}(f^A(Z))$ is convex about A. We can then solve the convex optimization problem

$$\widehat{A} \in \mathrm{argmin}_{\|A\|_* \leq R} \tilde{\mathcal{L}}(A)$$

where $\tilde{\mathcal{L}}(A) = \sum_{i=1}^{n} \mathcal{L}(f^A(x_n); y_n)$, n is the size of mini-batch, $\|\cdot\|_*$ denotes the nuclear norm, and R is a restriction. $\widehat{A}$ is then transformed to the corresponding parameters of the CNN.
As a result, the original non-convex problem is tranformed to a convex one.

3 The Construction of A Two-layer Convexified Convolutional Neural Network

A two-layer convolutional can be written as a function $f : \mathbb{R}^{d_0} \to \mathbb{R}^{d_2}$, it takes in a vector x, which is often the vector-representation of a picture. In the context of this report, the output $f(x)$ is a discrete distribution vector, i.e. the kth element $f_k(x) \in [0, 1]$ denotes the probability of x belonging to class k.
In a more common explanation, the input vector, or picture, x, is first separated to P *patches*, which can be written as a function $z_p(x) \in \mathbb{R}^{d_1}, 1 \leq p \leq P$.
Then, each patch is transformed to r scalars, which can be written as $h_j(z_p) = \sigma(w_j^T z_p), 1 \leq j \leq r$, where $w_j \in \mathbb{R}^{d_1}$, $\sigma : \mathbb{R} \to \mathbb{R}$ is in general a non-linear function. Each h_j is known as a *filter*.
Now we have $P \times r$ scalars. These scalars are finally summed together with weights, denoting as $\alpha_{k,j,p}$. The two-layer CNN can then be written as

$$f_k(x) := \sum_{j=1}^{r} \sum_{p=1}^{P} \alpha_{k,j,p} h_j(z_p(x)).$$

[1]Convexified Convolutional Neural Networks, see `https://arxiv.org/abs/1609.01000`.

When σ is identity, i.e. $\sigma(x) = x, x \in \mathbb{R}$, we can separate the trainable parameters α, w with other constants.