Vision: A Programming Approach

Xiyu Zhai, Jiang Xin, Jian Qian, Haochuan Li, Xiao Ma, Sasha Rakhlin, Piotr Indyk

Contents

Abstract

We introduce a fundamentally novel computer vision methodology that can explore possibilities beyond traditional computer vision and end-to-end deep learning. Using just CPUs, it can develop prediction programs with equal or better accuracy, efficiency and robustness, and full explainability. The effective computation needed in training is basically ignorable as opposed to that of deep learning.

1 Introduction

The field of computer vision has witnessed a lot of success primarily powered by deep learning. But what we get is still far from ideal.

The argument is as follows: there exists statistical and computational difficulty in prediction in pattern recognition tasks in computer vision, but deep learning attempts to address both classes of problems using only statistical function fitting.

So inevitably, deep learning leads to overparametrized model, slow to predict and costly to train and demand a lot of data and is not robust due to not incorporating domain specific data.

For analysis of MNIST, see later sections.

2 A New Paradigm for Machine Learning

The typical setup of machine learning theory states everything in mathematical terms. However, for many machine learning problem, especially those involved with pattern recognition, even the prediction(inference) involves nontrivial computation.

So the difficulty is twofold: statistical and computational.

What traditional machine learning and deep learning do is to solve problems of two different nature using statistical function fitting. This works in practice, but the model is slow, large, not robust, inexplainable, costly to train.

We claim there are far better approaches.

This section describe theoretically a new paradigm. We will explain how we reformulate machine learning setup in a way so that computational structure is captured. We will also present our solutions to the newly formulated problems, which combines the good of both the programming world and the machine learning world.

The following sections will be about how it works in practice.

2.1 Computational Difficulties in Machine Learning

The typical setup of machine learning is like:

first you have an input space $\mathcal{X}$ and output space $\mathcal{Y}$, and a set of functions $\mathcal{H}$ from $\mathcal{X}$ to $\mathcal{Y}$. There is an unknown function $f_0 : \mathcal{X} \to \mathcal{Y}$. There is a distribution $\mathcal{P}$ over $\mathcal{X}$, and i.i.d samples x_i with $y_i = f_0(x_i)$ according to $\mathcal{P}$. Then the problem is to choose an element in $\mathcal{H}$ that best approximates this unknown function f_0.

The problem with this formulation is that everything is stated in mathematically terms without mentioning its computational information.

The core concern is that functions in $\mathcal{H}$ might not be a function easy to compute, i.e. for $f \in \mathcal{H}$, it might take significant to compute for a given x the value of $f(x)$.

Example (Digit One)**.**

Example (Digit Zero)**.** Consider the loop space.

Example (Characters in General)**.** Consider the loop space.

2.2 Rethinking Hypothesis Space

2.3 Rethinking Training

3 Hypothesis Space

4 Training

5 Design of the Husky Programming Language

This deserves a separate long paper. But for the sake of logic completeness, we give a brief overview here.

The language itself will be general purpose. But here we focus on its usage in computer vision.

6 Detailed Comparison with Deep Learning

7 Future

7.1 Other Tasks in Computer Vision

One thing great about Husky is that it inherently does segmentation and detection even during classification tasks. So it would be easy to do them.

The sketching algorithm generalizes easily to 3D, resulting in fundamentally novel shape analysis. So 3D tasks are doable too.

7.2 Go Beyond Computer Vision

8 Contributions

This section is for listing individual contributions.

Xiyu Zhai's contribution can be summarized as

- line sketching algorithm and convex concave components and proof of its stability
- theoretical formulation of the computational difficulty in prediction
- the perspective of field vs particle
- type theory for features
- lazy feature computation
- most of the design, implementation and maintenance of the Husky programming language
- mnist, imagenet experiments
- realization that decision list is better than decision tree

Jiang Xin's contribution can be summarized as

- help with mnist, imagenet experiments

Jian Qian's contribution can be summarized as

- clarify theoretical formulation

Haochuan Li's contribution can be summarized as

- clarify theoretical formulation

Xiao Ma: TBA.

Sasha Rakhlin provides insights from learning theory. Fundamental machine learning concepts still apply.

Piotr Indyk provides insights from applied algorithms. The computation is inspired by LSH in the early days.