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 achieved 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.

For example, suppose we have a machine learning problem whose prediction is to see whether a noisy text input contains a certain regex pattern. The regex is unknown.

So the statistical problem is to determine the regex and the computational problem will be to simplify the prediction computation. If one use transformers, there are no separate steps, there is only one fitting process.

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 Rethinking 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 class of functions from $\mathcal{X}$ to $\mathcal{Y}$. There is an unknown function $f_0 : \mathcal{X} \to \mathcal{Y}$

2.1 Rethinking Hypothesis Space

2.2 Rethinking Training

3 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.

4 Detailed Comparison with Deep Learning

5 Future

5.1 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.

5.2 Go Beyond Computer Vision

6 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.