Practical advices for using ConvNets

Monday, June 11, 2018 11:02 PM

Using open-source implementation

Problem: how to replicate models by reading the papers.

- Look for open-source implementation.
  From Github

For instance: https://github.com/KaimingHe/deep-residual-networks impltmentation of Resnet, by the author uses the caffe framework.

Advantage: can use others' pre-trained model. and do transfer learning

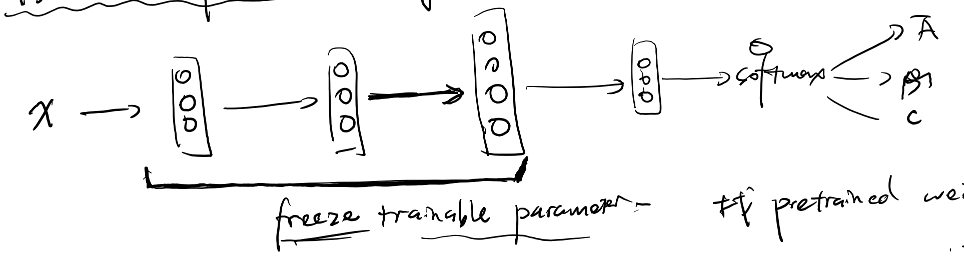
Transfer Learning

use pre-trained model ⟹ use it on your own task

Training is resource consuming.
  can download open-sourced weights ⟹ use on your own problem

Ex cat detection

Download pre-trained weights



找 pretrained weight，不让下降去 train 它们]
加速.

save to disk the first part of the model

- If your dataset is larger you can freeze fewer layer (more ~~trainable~~ parameters).
- To an extreme, ~~to~~ use the pretrained weights to initialize
  then train / update ALL of them

In practice, you will do MUCH BETTER using weights pretrained by others on large dataset

Data augmentation:

Having more data will always help. for the majority of CV tasks

Mirror / cropping / rotation / shearing / local

Color shifting. (play with RGB three channels +50, 0, -10 之类的.

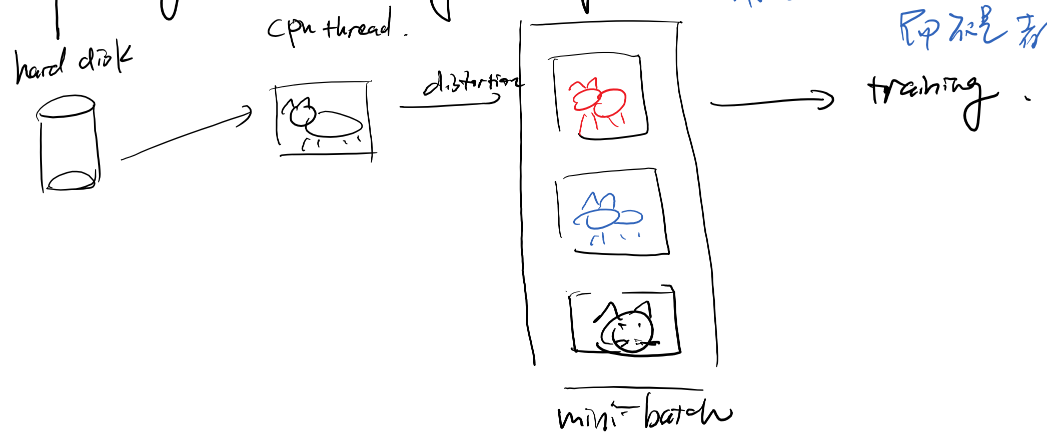
⇒ learning algorithm more robust to change of color

Advanced different ways to sample RGB colors. Implemented in AlexNet

PCA color augmentation: If your image is mainly purple.
⇒ add & subtract alot of blue & red. But leave G alone

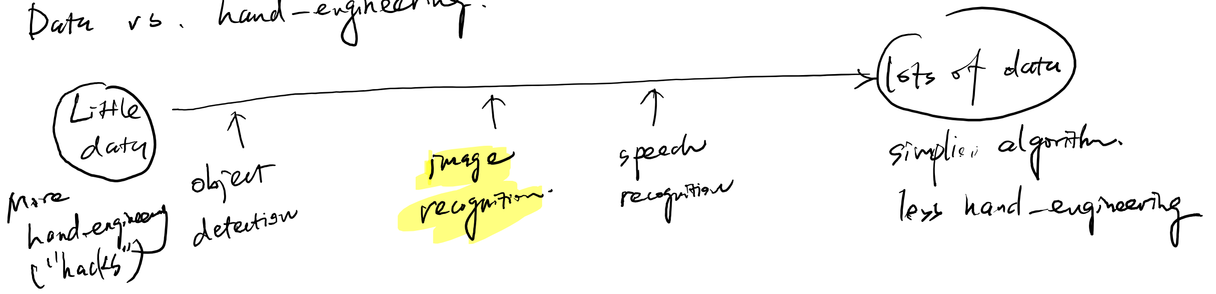
Implementing distortion during training

用独立的 cpu/GPU 来做 distortion (即时
即不是都存起来).



State of computer Vision

Data vs. hand-engineering.



Two sources of data

- Labeled data
- Hand engineered features / network architecture / other component.

· Hand engineered features/network [illegible]/ [illegible]

~~th~~ the fact that we don't have much data leads us to more complex network archetect-

When you have lots of labeled data
→ Better off spending time on building good sys. instead of hand-enginee[ring]

When you have little data — one thing that helps is TRANSFER LEARNING.

Tips for Doing well on Benchmark / winning competition

Ensembling

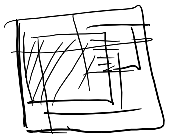
3-15 networks

- Train several NN independently, & average outputs.
  [not averaging weights].
  — time consuming! so not a good idea for production. (unless you have a generous computational budget.)

Multi-crop for test time. [crop images]

- Run classifier on multiple versions of test images and average results

~~在~~ 每幅 test image 随机 crop



Use open source code

- use architectures of networks published in the lit
- use open source implementation.
- ~~Fa~~ use pre-trained model and fine-tune with your datasets