

# Practical advices for using ConvNets

Wednesday, September 9, 2020 11:53 AM

## ① Using Open-Source Implementation

- Many NNs are difficult to replicate because of missing details or learning decay, parameter tuning, etc.
- Many researchers open source their code
- Downloading a network along with its weights can save a lot of time and compute to get started.

## ② Transfer Learning

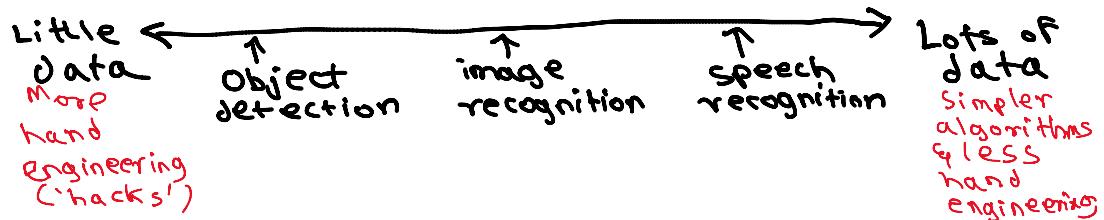
- If we are using a specific NN architecture that has been trained before, you can use this pretrained parameters/weights
- Pretrained models are often trained on large datasets and have been optimised well.
- Using them can save a lot of time and improve performance greatly

## ③ Data Augmentation

- Common augmentation methods in computer vision
  - mirroring
  - random cropping } mostly used
  - Rotation
  - Shearing
  - Local warping
  - Color shifting
- Implementing distortions during training
  - use another CPU thread to apply distortions to form mini-batches of data that are passed on to training
- Data augmentation also involves hyperparameters like amount of rotation / color shift, etc

## ④ State of computer vision

### • Data vs Hand-engineering



- Two sources of knowledge
  - Labeled data
  - Hand engineered features/network architecture /other components ↳ requires insight
- Tips for doing well on benchmarks/ winning competitions
  - Ensembling
    - takes time ↳ Train several networks independently and average their outputs
    - Multi-crop at test time
      - uses more memory ↳ Run classifier on multiple versions of test images and average results
  - use open source code
    - use architectures of networks published in the literature
    - use open source implementations if possible
    - use pre trained models and fine tune on your dataset