Deep Learning Content School
(ML Production)
Day 1

@philipshurpik

reface

What to expect :)

- NN Performance, measurement, speed optimization
- High performance python engineering (multiprocessing, GIL, shared memory)
- Video processing - codecs, libraries, etc.
- Create pipeline with NNs from scratch (MapReduce, python processes, queues...)
- Recent model serving frameworks (BentoML, TorchServe...)
- Deploy to cloud

NN Performance (day 1)

reface

NN Model size

- Model size (file size)
  - matters especially for mobile devices
  - contributes to model GPU memory consumption
- What can be done (simple solution)?
  - save weights in format they will be used (half, uint8…)
  - use different NN compilers, export formats (Torschscript, ONNX…)

reface

NN GPU memory consumption

- During initialization
  - Even tensor with one element consumes memory # try *torch.tensor([0]).cuda()*
  - Depends on GPU architecture (CUDA capabilities, cuda and torch version)
- During inference
  - Batch size
  - Precision (Float, half)
- Why does it matters?
  - How many models can be run in one process
  - How many cuda processes can be run on one GPU
- Demo ^_^

NN inference time

- Precision (float, half)
- Batch size (1, 2, 4, 6…)
    - + higher speed
    - - harder to write code (aggregation)
    - - bigger latency if aggregate different requests
    - - bigger memory consumption

NN inference time

- Preprocessing
  - resize (on cpu or on gpu ?)
  - prepare data
- Moving data CPU2GPU, GPU2CPU
  - torch.cuda.synchronize()
- Postprocessing (nms, keypoint heatmaps...)
  - python code is not efficient (one thread, cycles ...)
  - batching in post processing is recommended
  - c++ for hard cases
  - look in torchvision.ops (https://pytorch.org/vision/stable/ops.html)
- Demo ^_^