Tuning the Performance of Convolutional Neural Network for Image Classification on GPU

Alibaba Group

Agenda

- Adoptions of Image classification or image recognition at Alibaba
- Easy ways to improve performance of Caffe
- Further performance optimization of convolution layer
- Ongoing works

Image classification at Alibaba

- Product Display Classification
- Fashion Style Classification
- Buy-by-photo mobile app, search for visually similar products by images
- Leverage Caffe framework

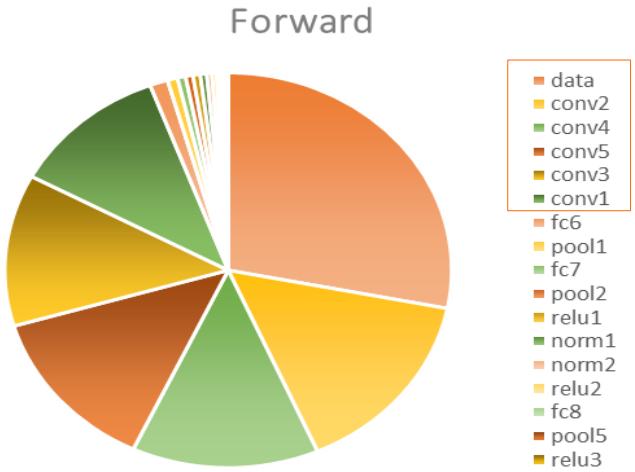
Model-Upper/Item-Bottom/Multi-Object

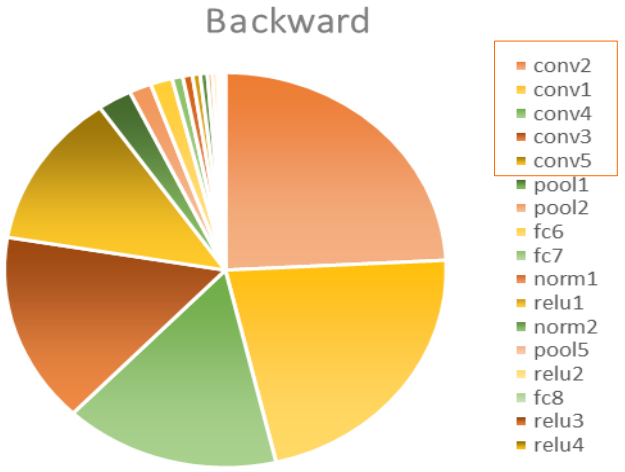
Sweet / Street / Office

Alibaba Group

Profiling Caffe

- Most expensive part

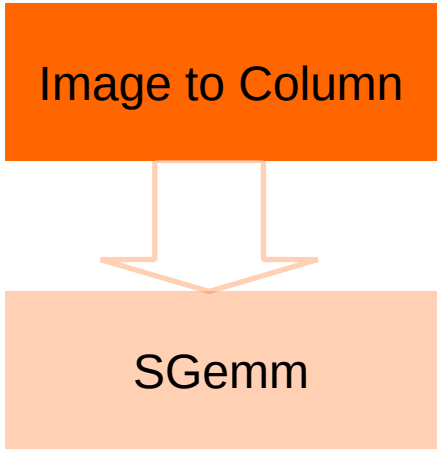




Caffe spends more than 70% of time on Convolution layers !

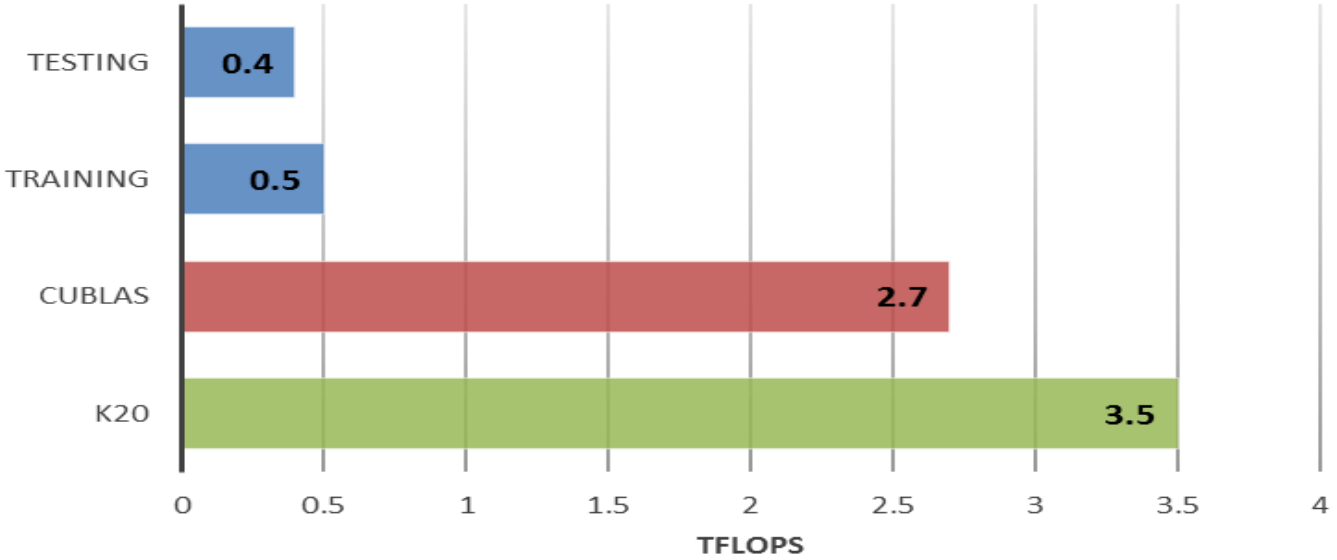
Convolution layer

- How does the convolution layer work in Caffe



The gap

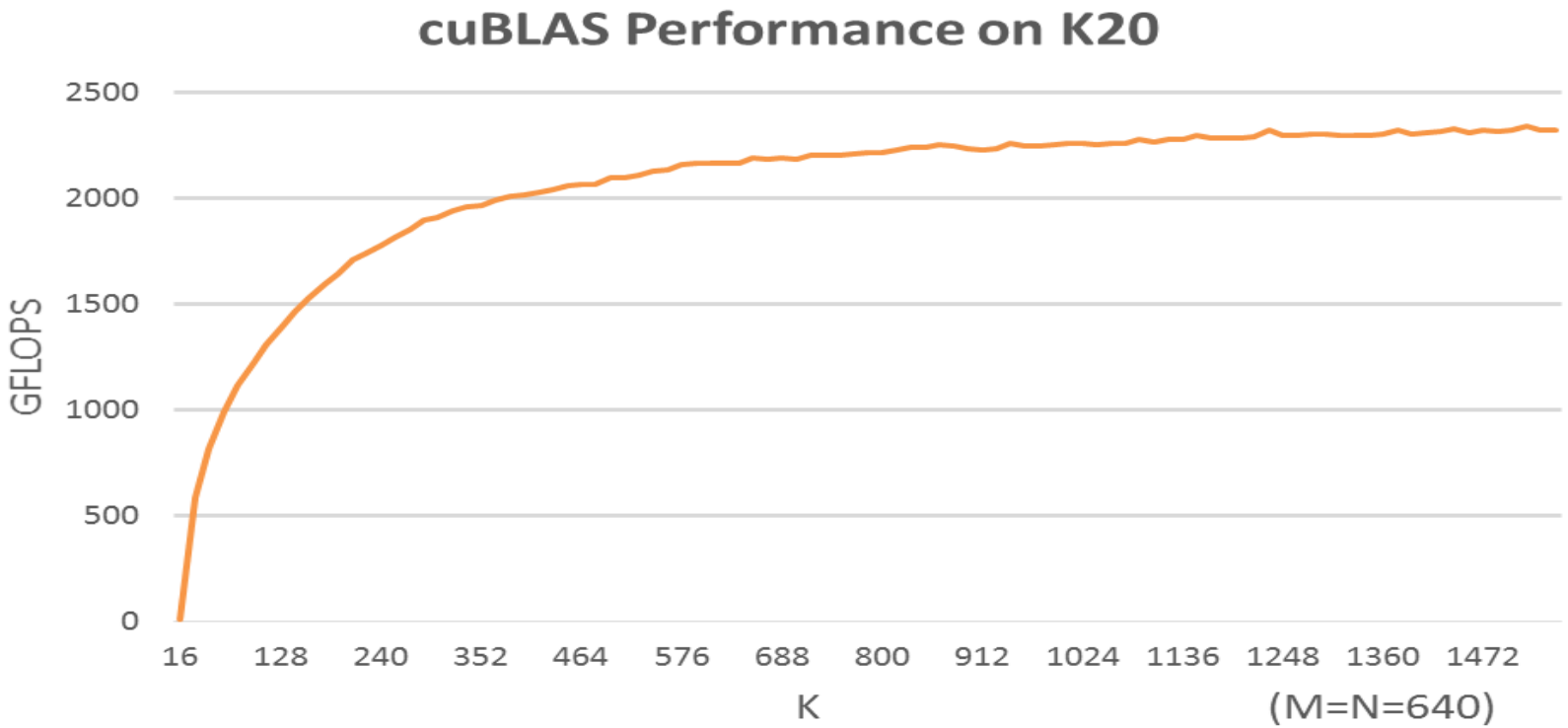
- Is it really fast?



Blue:
Caffe(imagenet model)
Red:
Sgemm routine of Cublas
Green:
Peak of K20

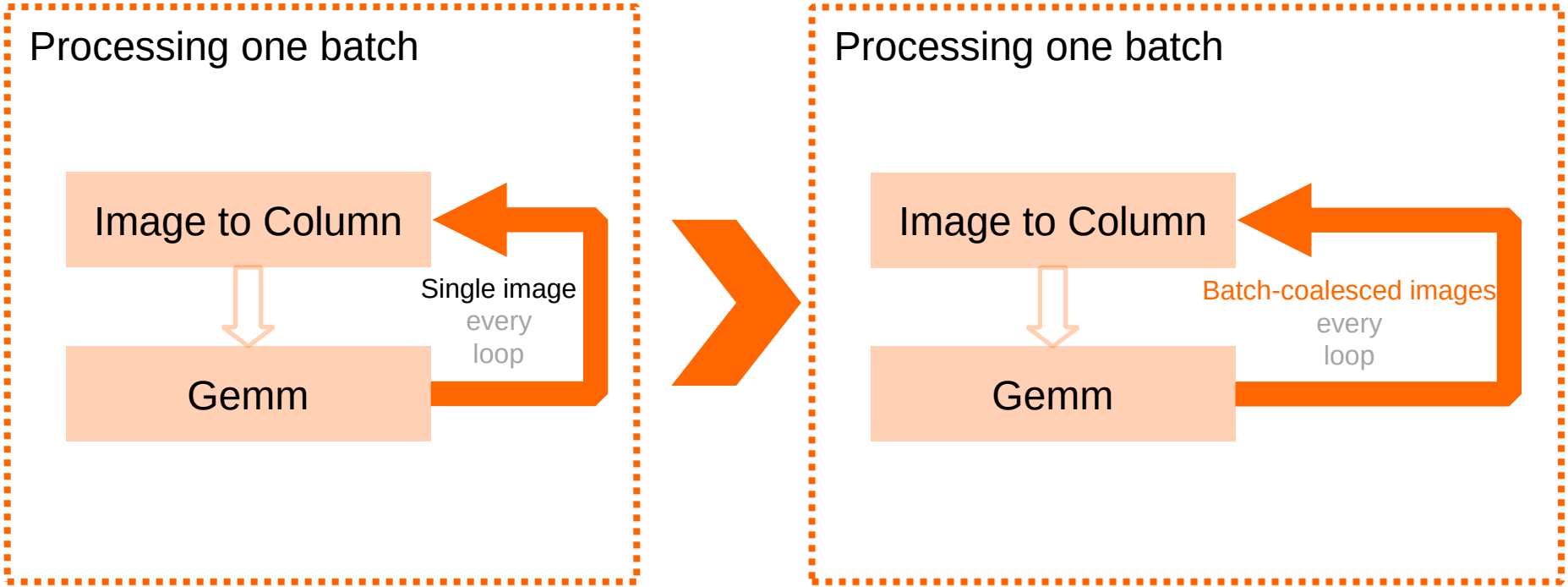
ImageNet model, refer to the ILSVRC12 challenge

How does Cublas Sgemm perform



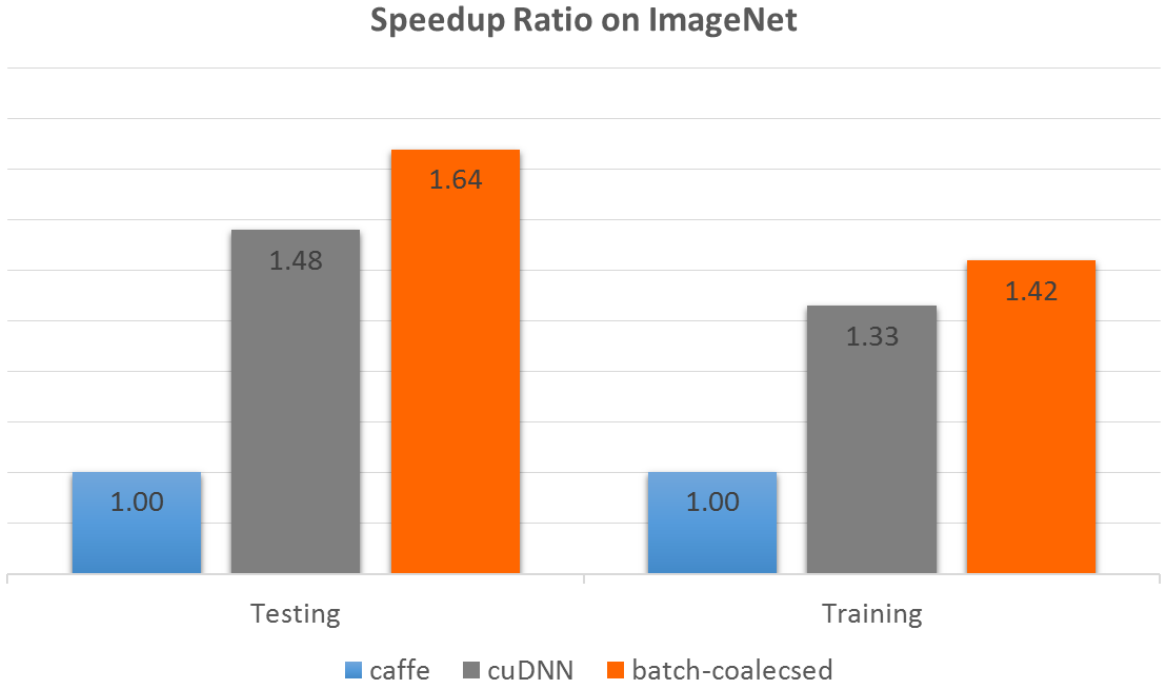
Easiest way to narrow the gap

- To Overcome the low efficient of SGEMM at small scale



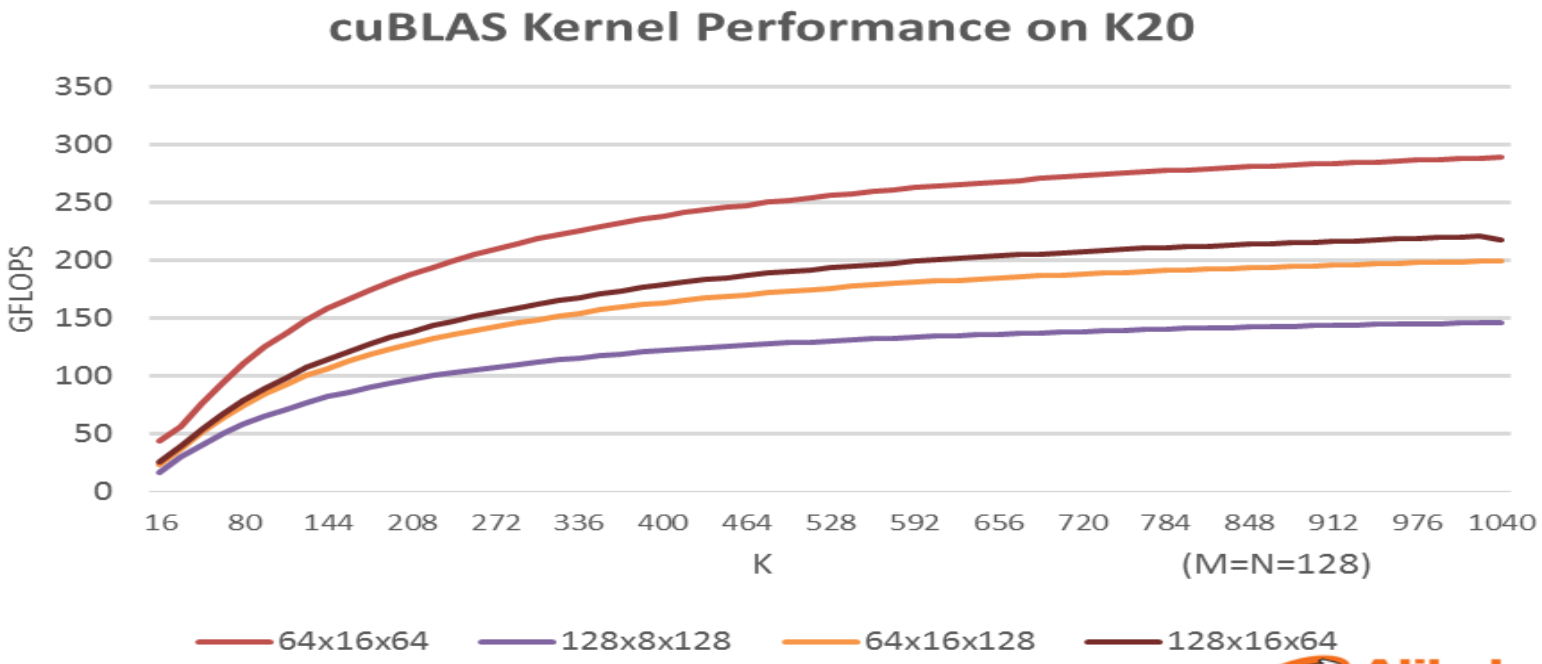
Performance of Fast mode

- Titan black, mini-batch size is 256



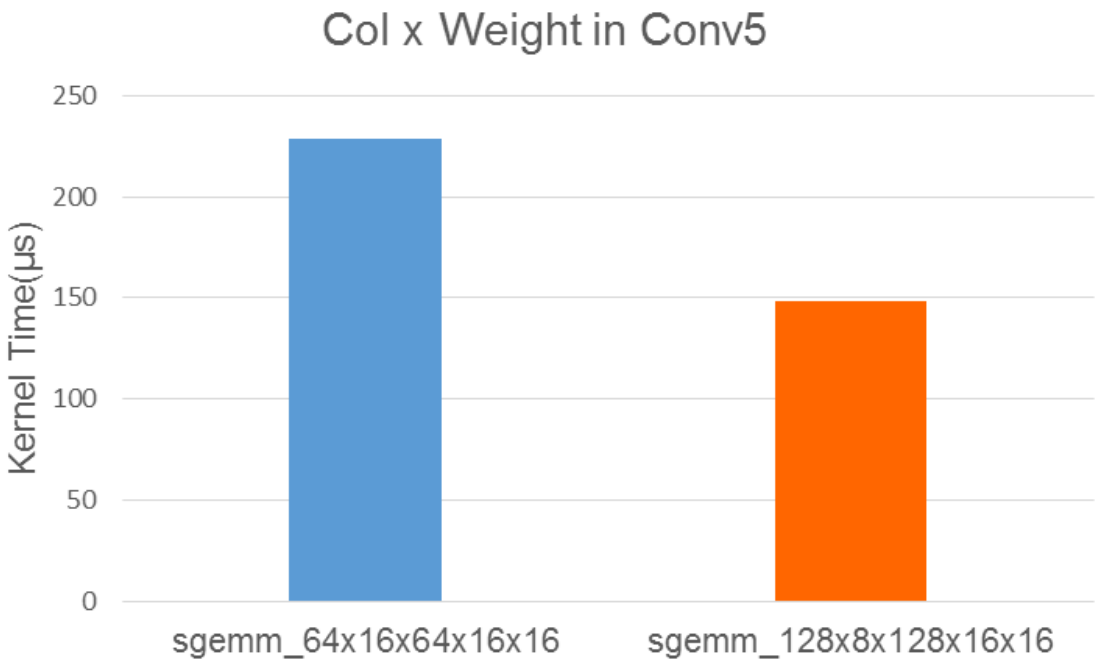
Moving forward

- How is cublas sgemm implemented



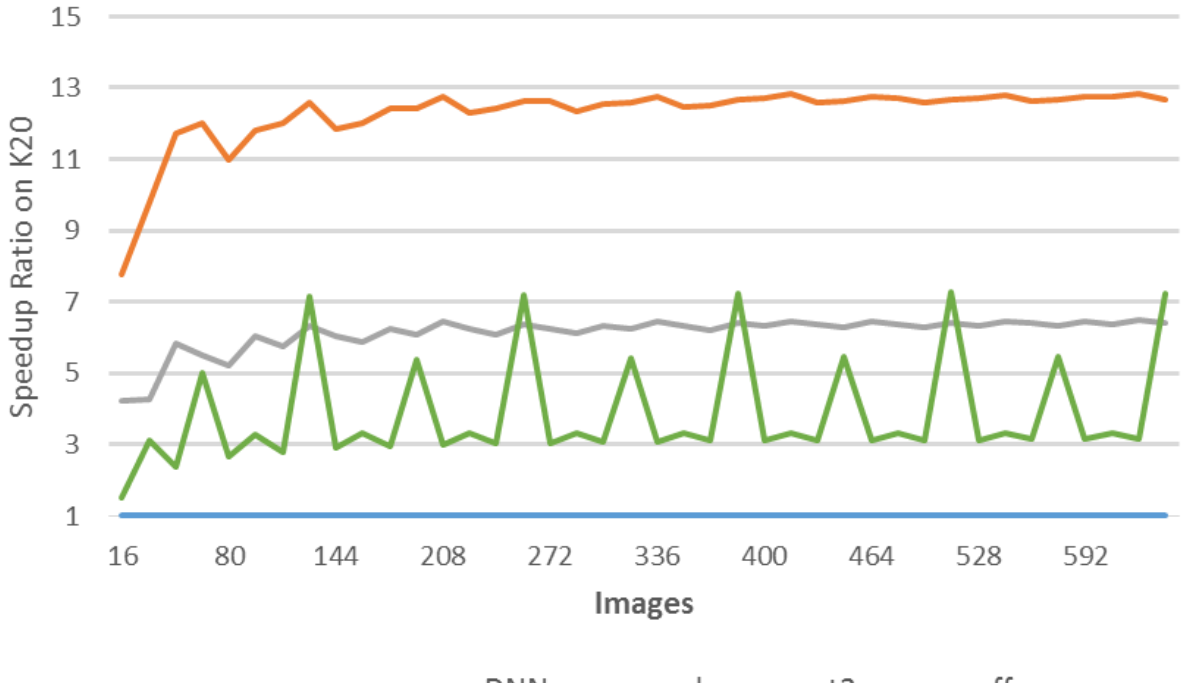
Use high performance sgemm routines

- Example: ImageNet convolution layer “conv5”: M = 96, N=3025, K=363
- cuBLAS use: sgemm_64x16x64x16x16, slow!
- We use: sgemm_128x8x128x16x16 to get the same result, 1.54x faster on K20 !



Implement our own conv layer

- Auto-gen gpu kernels for convolution layers
- Kernels are implemented in PTX assembly



Conv2 from Alex's Net, Height = 16; Width = 16; Channel = 5; Stride = 1; Ksize = 5; Pad = 2; Neuron = 32

Is PTXAS good enough?

- Problem
  - Register usage
  - Manipulate “control code” on Kepler
- Our own assembler for Kepler
  - Probe native ins
  - Probe control ins
  - Ongoing
- **Some users need a native assembler, please!**

snippet of instructions from sgemm kernel, sm_35

```
/* 0x090010101c1c101c1c */
FFMA R23, R83, R84, R23;
FFMA R33, R88, R84, R33;
FFMA R36, R88, R85, R36;
NOP;
FFMA R45, R89, R84, R45;
FFMA R32, R89, R85, R32;
NOP;
/* 0x0880101410141014 */
FFMA R5, R80, R86, R5;
FFMA R2, R81, R86, R2;
FFMA R14, R81, R87, R14;
FFMA R7, R80, R92, R7;
FFMA R3, R80, R87, R3;
FFMA R8, R81, R92, R8;
NOP;
```

Other ongoing works

- Convert model from Single-precision floating points to
  - half-precision (maxwell)
  - flexible fixed-points (FPGA)

Thank You

- Download the mobile app at taobao.com and try out Buy-by-Photo