深度学习框架内存优化研究

Ping Chen

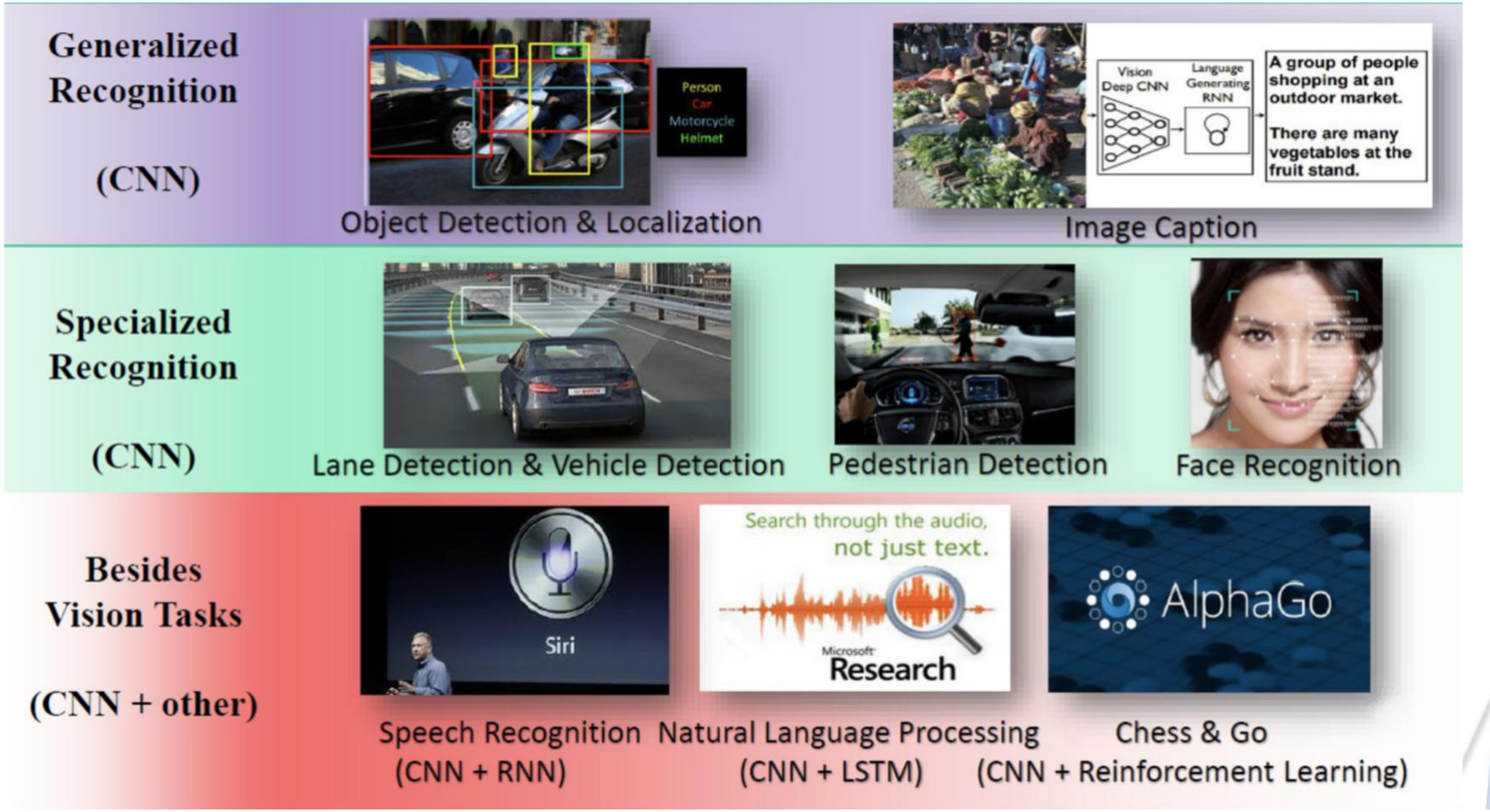
2021/5/9

Zhejiang University

Outline

- **深度学习背景**
- 内存交换
- 重计算
- 压缩技术

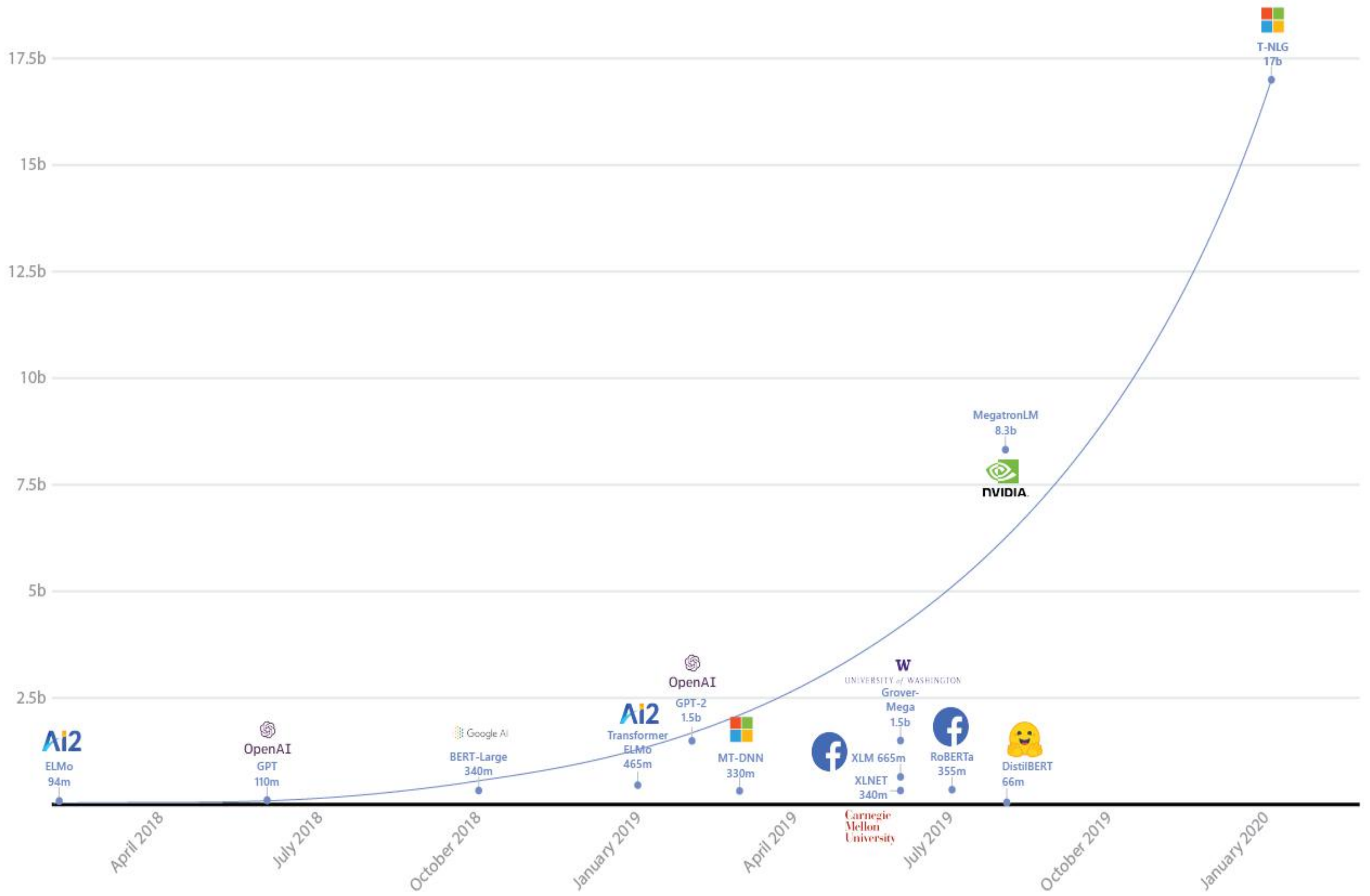
深度学习给社会带来的机遇



深度学习在自动驾驶、人脸识别、自然语言处理、博弈等方面取得了巨大的成果。

深度学习的发展趋势

模型训练时需要大量的显存空间:



- InceptionV4设置batch size为32训练ImageNet需要 40GB显存空间[1];
- BERT拥有768个隐藏层，在Batch size设置为64时需要73GB的显存空间[2];
- 使用ImageNet训练Wide ResNet-152，并设置Batch size为64需要显存180GB[3];

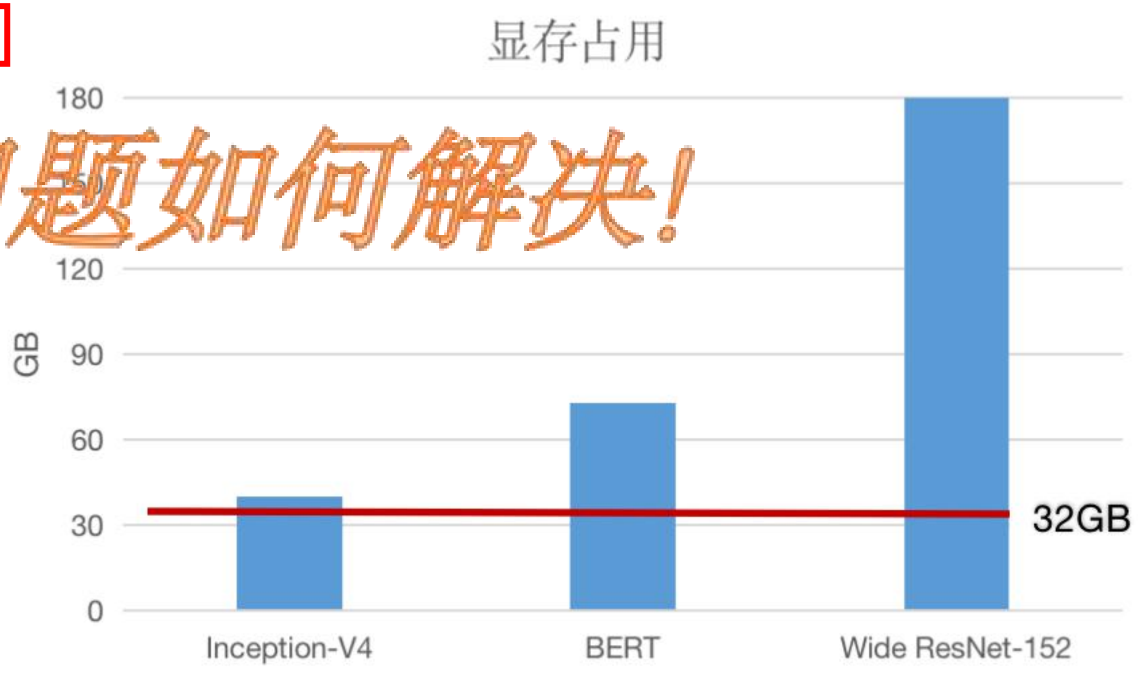
深度学习模型的参数量随着发展呈现出指数增长趋势，图中表明在2020年1月份的Turing Natural Language Generation (T-NLG)模型拥有170亿的参数量≈63GB内存

深度学习加速器现状

售价￥70000，价格昂贵！！！

指标 GPU	显存容量/GB	显存带宽/Gbps	Tensor Core	FP32 峰值/TFLOPS
V100(SXM2)	32 HBM2	900	640	15.7
TITAN RTX	24 GDDR6	672	576	16.3
P100(SXM2)	16 HBM2	732	NA	10.6
TITAN V	12 HBM2	652.8	640	15
RTX 2080Ti	11 GDDR6	616	544	13.4
RTX 2080	8 GDDR6	448	368	10.1
RTX 2070	8 GDDR6	448	288	7.5
TITAN Xp	12 GDDR5X	547.7	NA	12
RTX 1080Ti	11 GDDR5X	484	NA	11.3
TITAN X	12 GDDR5	336.5	NA	11
GTX 1080	8 GDDR5X	484	NA	8.9
RTX 1070Ti	8 GDDR5	256	NA	8.1
RTX 1070	8 GDDR5	256	NA	6.5
RTX 1060	6 GDDR5	256	NA	4.4

存储容量不足问题如何解决！



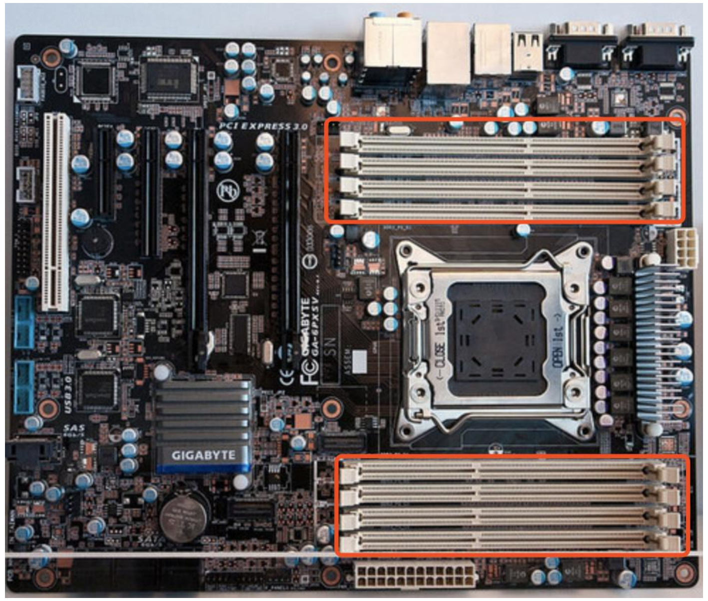
左图为NVIDIA公司生产的常用深度学习GPU性能指标，其中目前性能较高的V100最大容量仅为32GB；

右图表示：最大显存GPU（32GB）已经不能满足当前深度学习的训练需求；

Outline

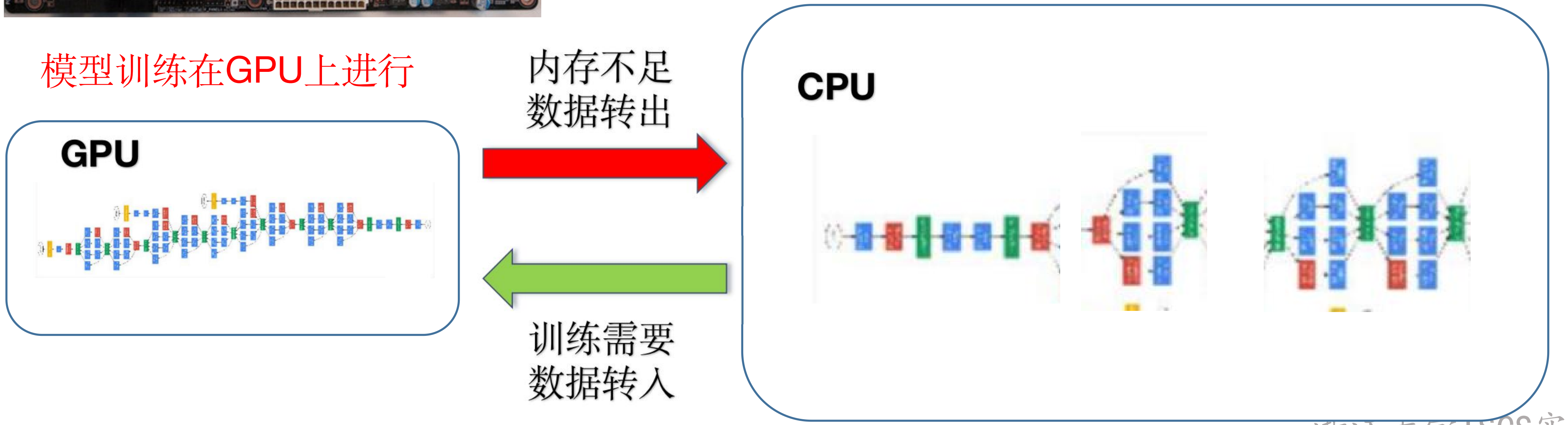
- 深度学习背景
- **内存交换**
- 重计算
- 压缩技术

数据交换方案

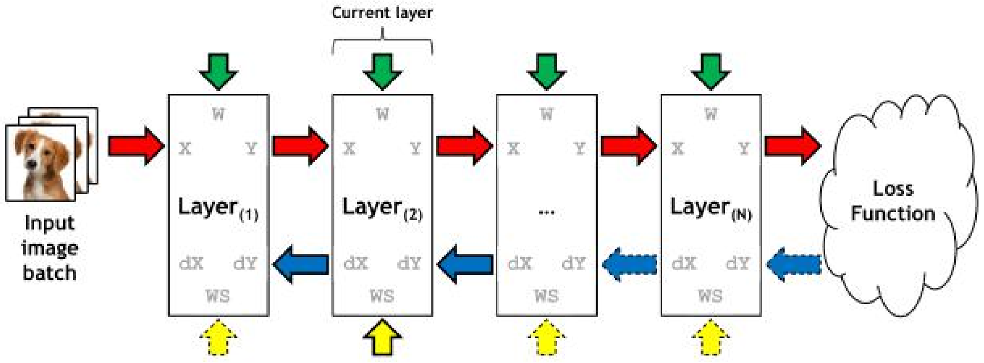


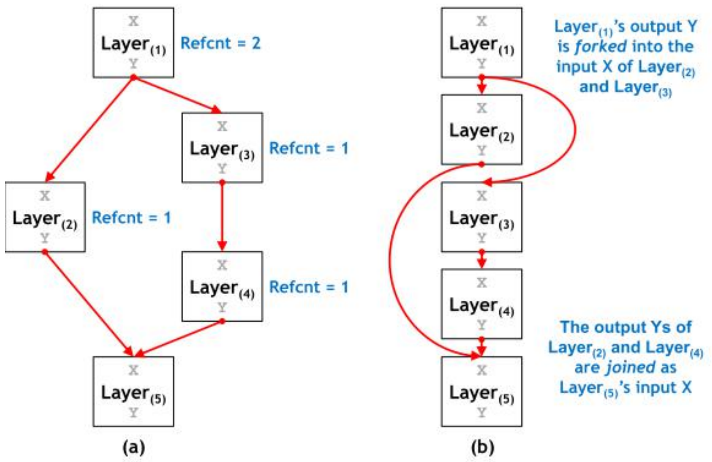
当今服务器可配置32GB*N的DRAM容量，远大于GPU显存；如何利用CPU DRAM与GPU DRAM异构系统设计新的内存优化方案已经成为研究热点。

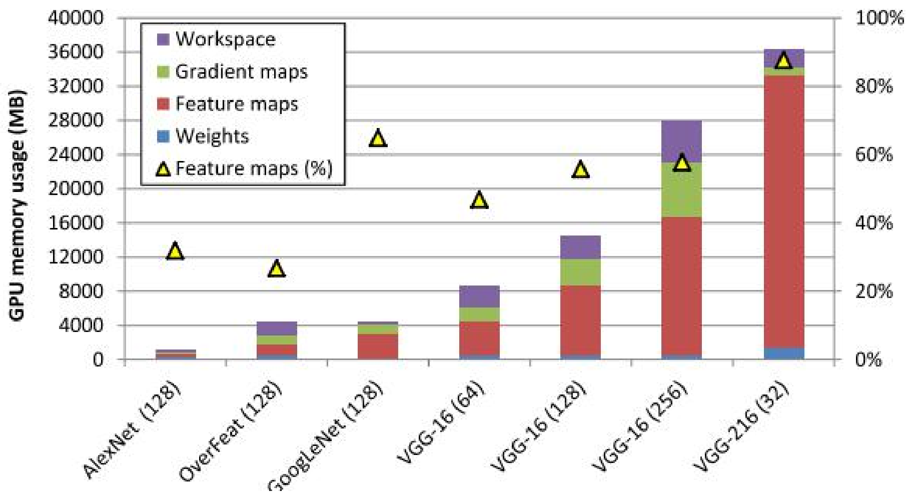
模型训练在GPU上进行



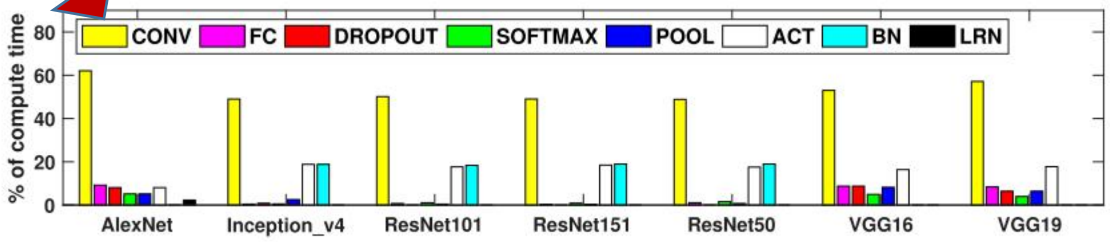
DNN数据的特征







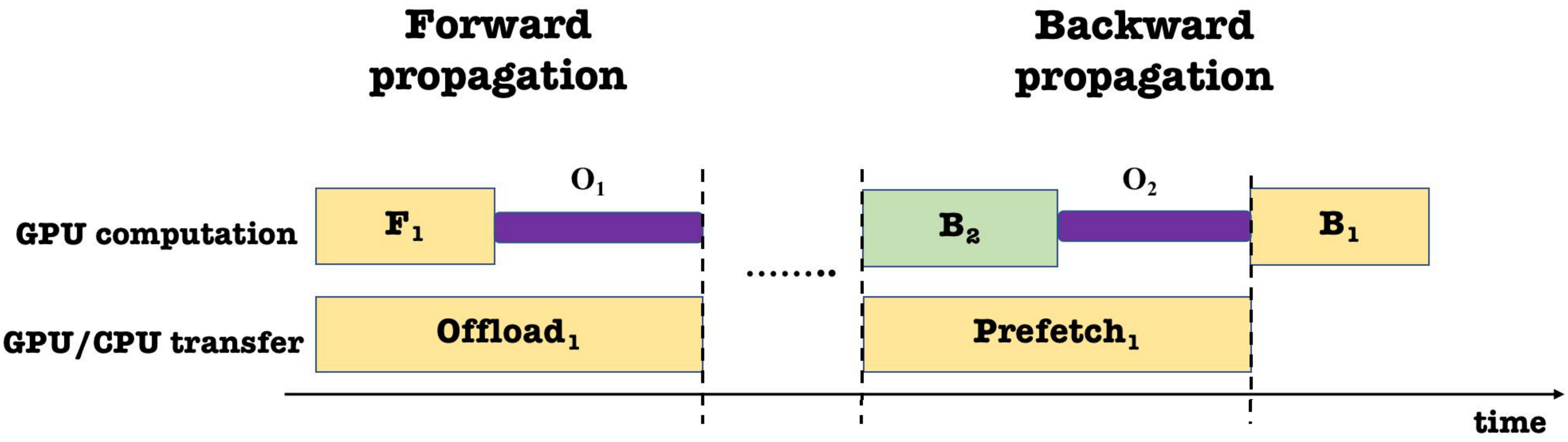
Feature map的空间占比非常高



(a) breakdown of execution time by layer types

Conv的计算时间相对较久

GPU-CPU转移方案-vDNN

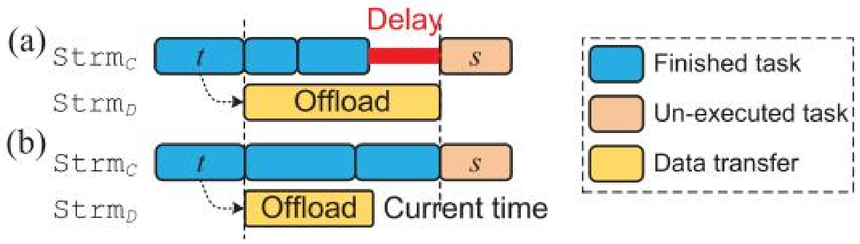


vDNN内存转移方案示意图

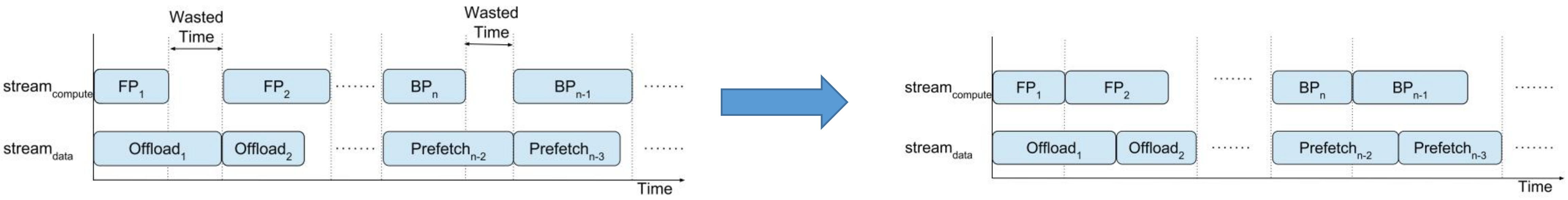
- 前向传播时选择卷积层的前一层的输出数据进行转出，反向传播将数据提前转移回来[1]；

[1] M. Rhu, N. Gimelshein, J. Clemons, A. Zulfiqar, and S. W. Keckler, “VDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design,”in Proceedings of the Annual International Symposium on Microarchitecture, MICRO, 2016, vol. 2016

GPU-CPU转移方案-其他



■1 moDNN[1]在解决了vDNN不足的基础上使用启发调度的思想选择**合适的CONV算法**（有快有慢，占用空间不同），达到内存与性能均优的情况；



■2 vDNN++在解决了vDNN转移模式不足的基础上同时设计新的显存分配模式降低碎片[2]；

■3 SwapAdvisor使用遗传算法、贝叶斯优化器等启发算法进行转移策略的搜索[3][4]；

[1] X. Chen, D. Z. Chen, and X. S. Hu, “MoDNN: Memory optimal DNN training on GPUs,” Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018, vol. 2018-Janua, pp. 13–18, 2018.
[2]S. B. Shriram, A. Garg, and P. Kulkarni, “Dynamic memory management for GPU-based training of deep neural networks,” Proceedings - 2019 IEEE 33rd International Parallel and Distributed Processing Symposium, IPDPS 2019, pp. 200–209, 2019.
[3] C. C. Huang, G. Jin, and J. Li, “SwapAdvisor: Pushing deep learning beyond the GPU memory limit via smart swapping,” in International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, 2020, pp. 1341–1355.
[4] Efficient Memory Management for GPU-based Deep Learning Systems arXiv 2019

GPU-CPU转移方案-相关文章

- [1] M. Hildebrand, J. Khan, S. Trika, J. Lowe-Power, and V. Akella, “AutOTM: Automatic tensor movement in heterogeneous memory systems using integer linear programming,” in International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, 2020, pp. 875–890.

- [2] J. Ren, J. Luo, K. Wu, M. Zhang, and D. Li, “Sentinel: Runtime Data Management on Heterogeneous Main MemorySystems for Deep Learning,” 2019.

- [3] D. Yang and D. Cheng, “Efficient GPU Memory Management for Nonlinear DNNs,” HPDC 2020 - Proceedings of the 29th International Symposium on High-Performance Parallel and Distributed Computing, pp. 185–196, 2020.

转移方案的不足

- 1 转移带宽受限（PCIe有限的带宽）；
- 2 不同层的特征不同（计算时间、中间数据大小等），导致转移方案并不高效；

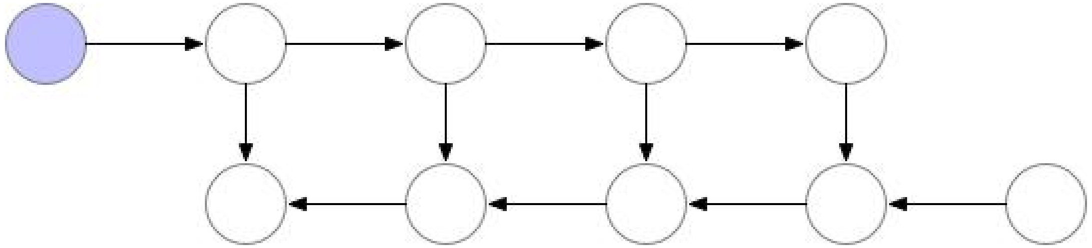
需要更为高效的整体方案!

Outline

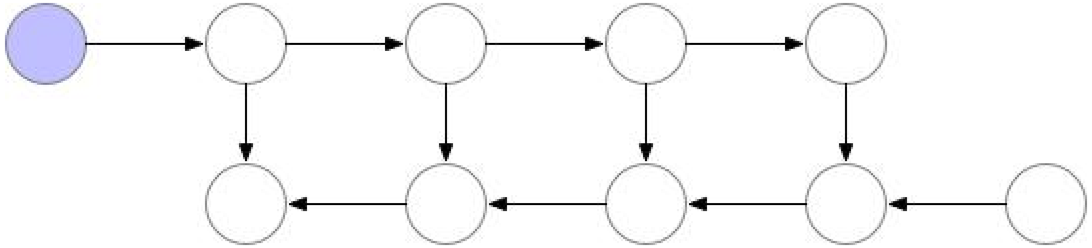
- 深度学习背景
- 内存交换
- **重计算**
- 压缩技术

Gradient Checkpointing[1]（重计算）

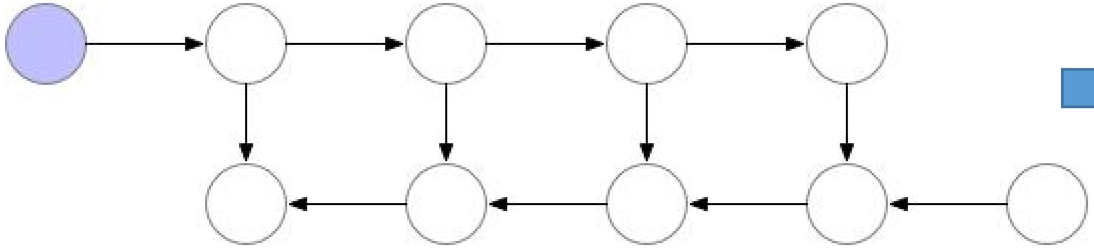
1.将用到的数据全都放在显存中



2.只将当前需要用到的数据放在显存中



1，2做法的折衷，花费一部分显存存储部分中间数据，加快计算[1]



■Checkmate[2]：用线性整数规划搜索最优的策略；

（当层变多后，搜索效率低，实用性不强）

[1]T. Chen, B. Xu, C. Zhang, and C. Guestrin, “Training Deep Nets with Sublinear Memory Cost,” pp. 1–12, 2016.
[2]P. Jain et al., “Checkmate: Breaking the Memory Wall with Optimal Tensor Rematerialization,” arXiv preprint arXiv:1910.02653, 2019.

交换方案+重计算-SuperNeurons[1]

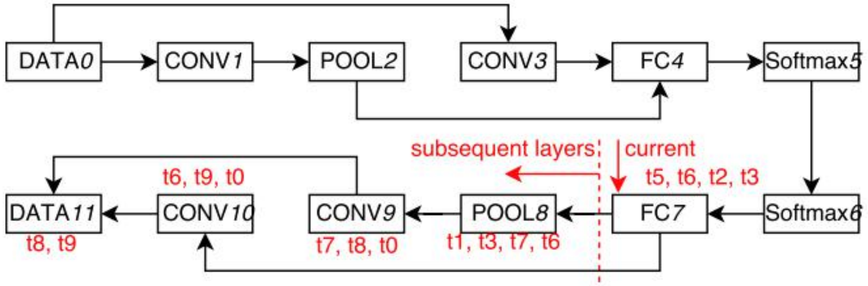


图1

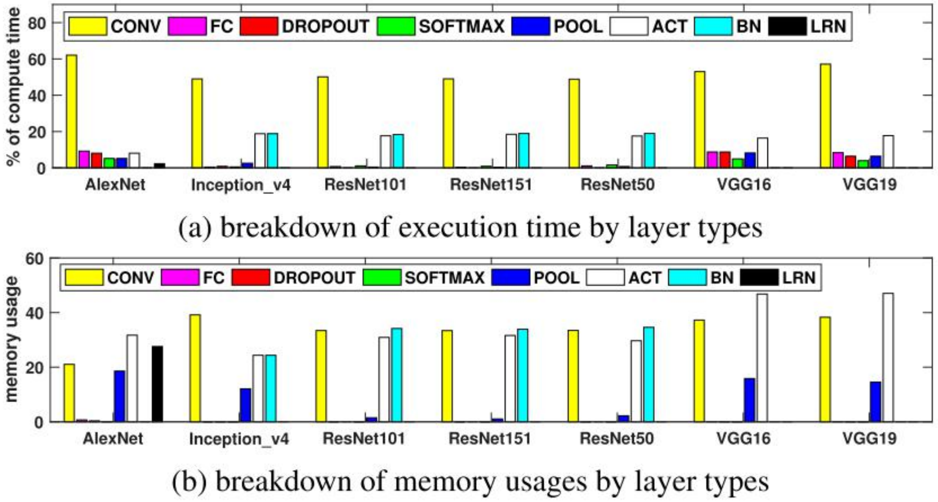


图2

1 在反向传播中逐渐释放不需要的Tensor（图1）；

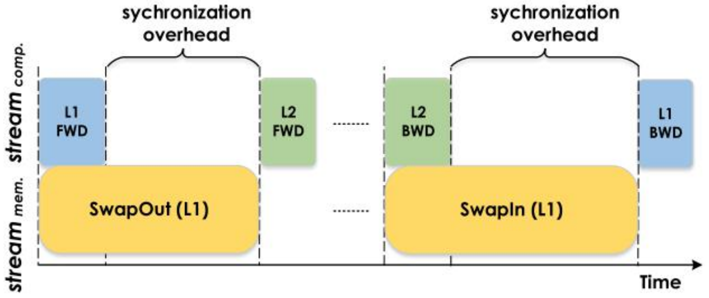
2 Conv计算时间长，不适合重计算，所以仅将Conv的输出进行转移（图2）；

3 POOL, ACT, LRN 以及BN层计算时间短，占用空间多，所以对这些层进行重计算（图2）；

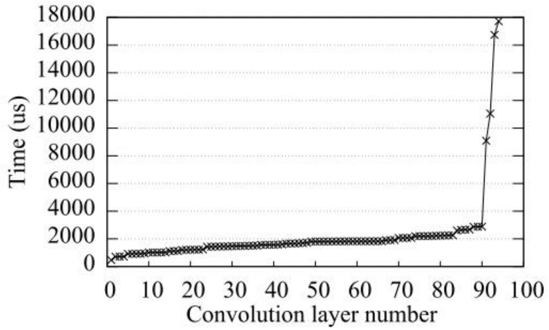
启发式的思想

[1]L. Wang et al., “SuperNeurons: Dynamic GPU memory management for training deep neural networks,” in Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPOPP, 2018, pp. 41–53.

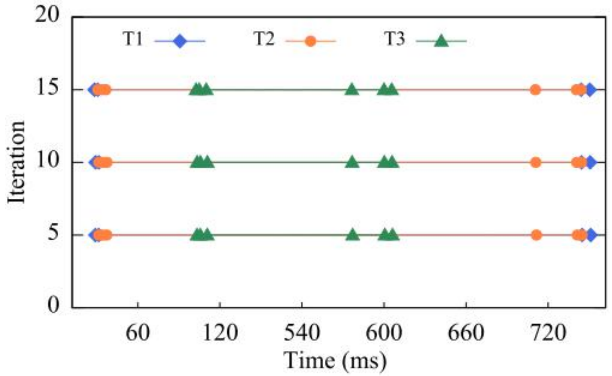
交换方案+重计算-Capuchin[1]



vDNN的不足



卷积执行时间差别较大，并不都是很久（InceptionV3）



较为规律的访问模式

在设计思路上前者都为**启发式**:
对某些层的优化**先入为主**

不能简单的仅将卷积
前面的数据进行转移

前面的数据**生命周期长**,
更值得优先被处理

[1]X. Peng et al., “Capuchin: Tensor-based GPU memory management for deep learning,” in International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, 2020, pp. 891–905.

交换方案+重计算-Capuchin

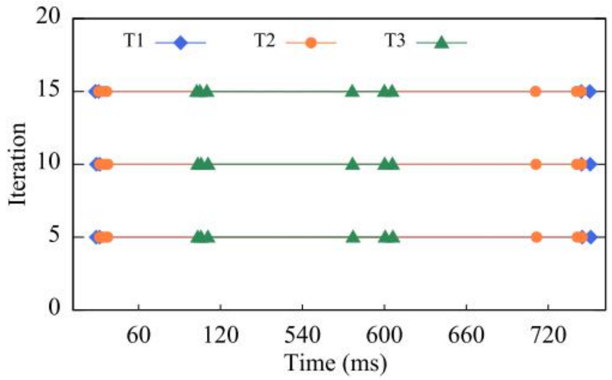
在设计思路上：不能为等待转移结束

不能简单的仅将卷积前面的数据进行转移

前面的数据**生命周期长，**更值得优先被处理

Capuchin-结合转移+重计算设计新的高效思路[1]

- ①由于转移可以隐藏在计算中，重计算不可避免的会引入额外开销，所以先选择转移的Tensor；即：先根据Tensor的寿命进行排序（可以理解为下图中线长的Tensor优先，左图）。
- ②对排序后的Tensor依次进行转移决策，选择转移开能够完全隐藏的Tensor；
- ③根据MSPS（右图）指标对重计算Tensor进行选择； **-- 保存的空间越大，重计算时间越小的Tensor更值得被重计算；**

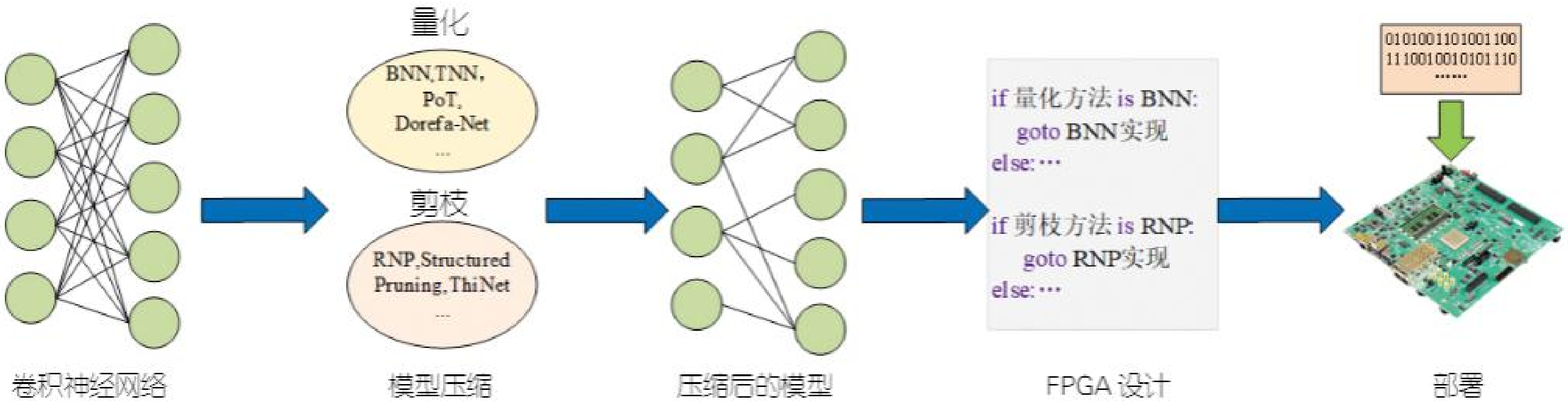


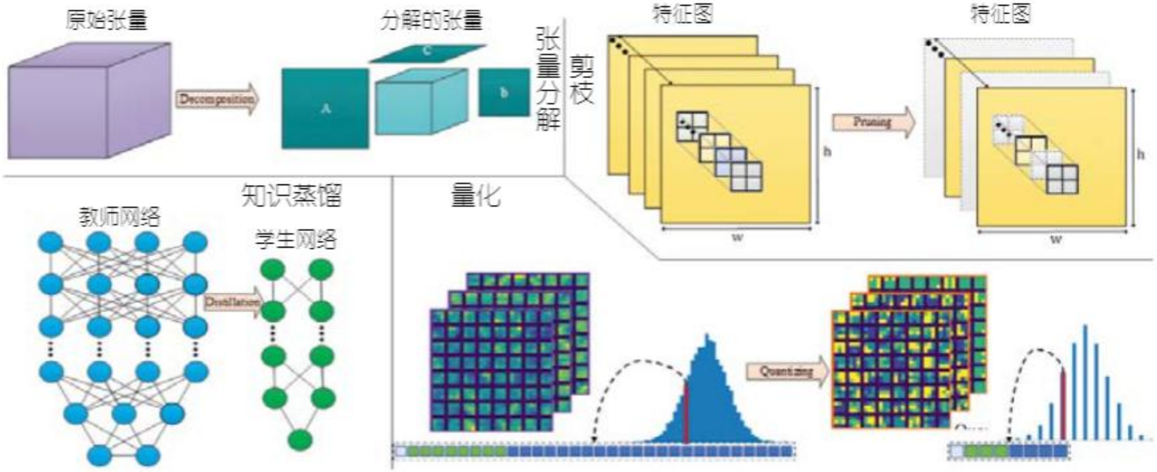
$$MSPS = \frac{Memory\ Saving}{Recomputation\ Time}$$

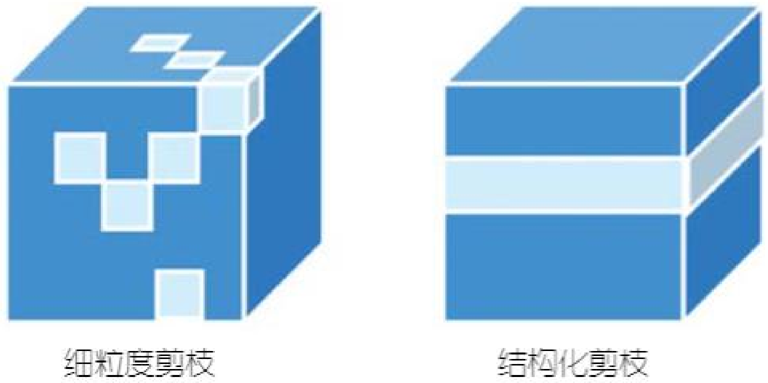
Outline

- 深度学习背景
- 内存交换
- 重计算
- **压缩技术**

量化、剪枝







量化：将数据聚类，并用某个数代表该类别的所有数；

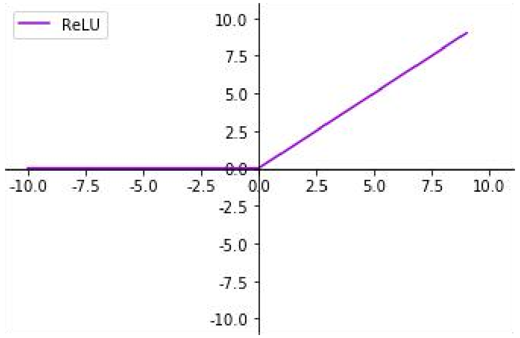
剪枝：减去部分参数值，并不过分损耗模型精度；

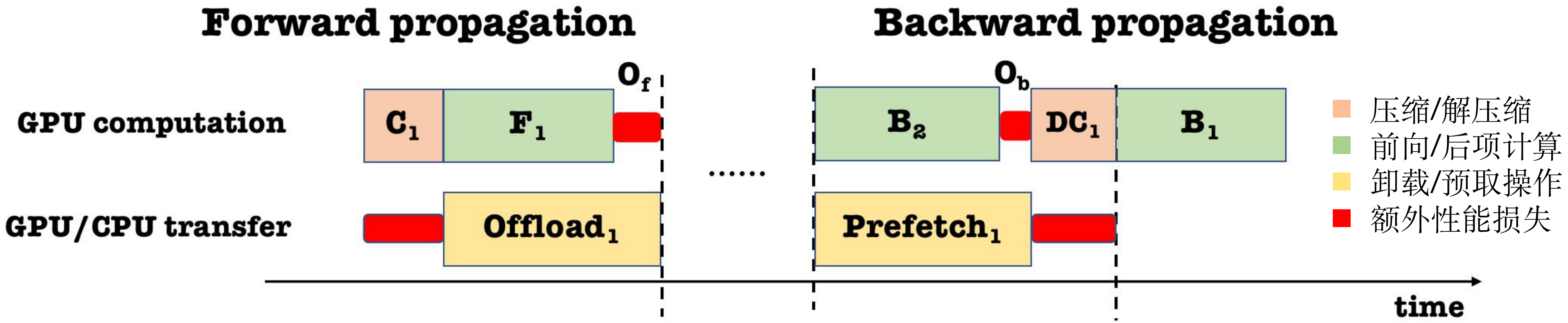
[1] https://dl.ccf.org.cn/reading.html?id=5354164101597184
[2]Y. Gong, L. Liu, M. Yang, and L. Bourdev, “Compressing Deep Convolutional Networks using Vector Quantization,” pp. 1–10, 2014.
[3]H. Li, H. Samet, A. Kadav, I. Durdanovic, and H. P. Graf, “Pruning filters for efficient convnets,” in 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings, 2019, no. 2016, pp. 1–13.

cDMA方案 - 缓解数据转移引入的额外性能开销

在GPU中增加**硬件**对稀疏数据进行**压缩**、**解压缩**，如下图。

- 机遇：ReLU为模型的输出带来了稀疏特性（ReLU输出数据含有大量的0）



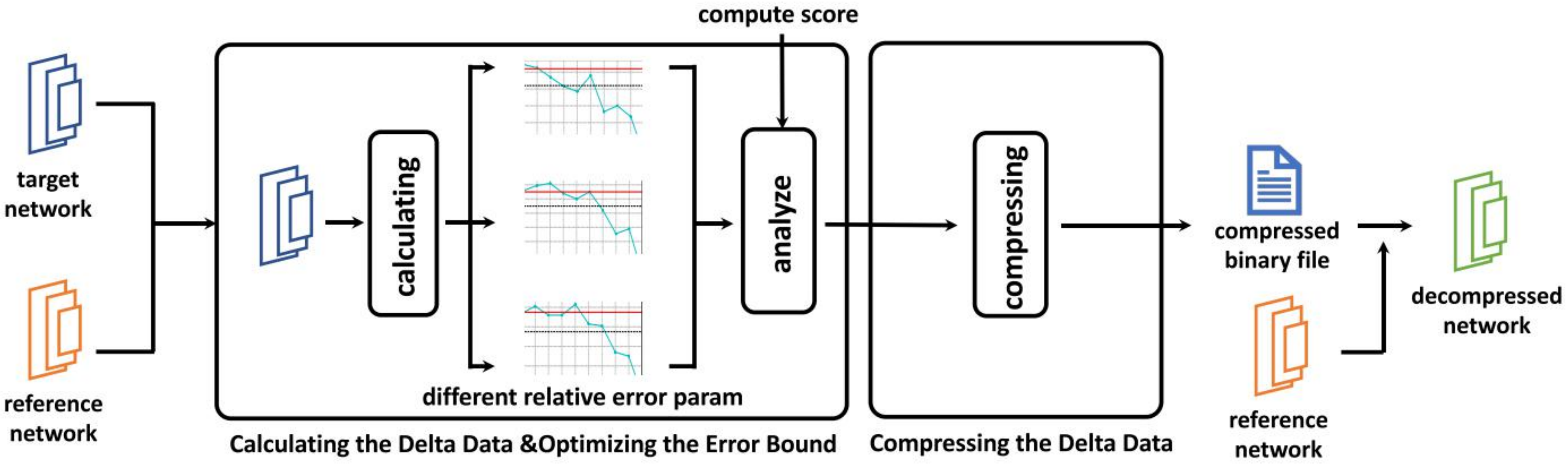


cDMA示意图

[1] M. Rhu, M. O'Connor, N. Chatterjee, J. Pool, Y. Kwon, and S. W. Keckler, "Compressing DMA Engine: Leveraging Activation Sparsity for Training Deep Neural Networks," Proc. - Int. Symp. High-Performance Comput. Archit., vol. 2018-Febru, pp. 78–91, 2018.

Delta-DNN 对weight进行有损压缩

对神经网络的Weight进行SZ有损压缩，降低Weight大小，从而加速CKPT保存与网络weight传输的过程；



[1]Z. Hu et al., “Delta-DNN: Efficiently Compressing Deep Neural Networks via Exploiting Floats Similarity,” ACM International Conference Proceeding Series, 2020.

压缩-相关文章

- [1] A. Jain, A. Phanishayee, J. Mars, L. Tang, and G. Pekhimenko, “GIST: Efficient data encoding for deep neural network training,” Proceedings - International Symposium on Computer Architecture, pp. 776–789, 2018.
- [2]B. Akin, Z. A. Chishti, and A. R. Alameldeen, “ZCOMP: Reducing DNN cross-layer memory footprint using vector extensions,” Proceedings of the Annual International Symposium on Microarchitecture, MICRO, pp. 126–138, 2019.
- [3]S. Jin, S. Di, X. Liang, J. Tian, D. Tao, and F. Cappello, “DeepSZ: A novel framework to compress deep neural networks by using error-bounded lossy compression,” HPDC 2019- Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing, pp. 159–170, 2019.

总结

一、转移

二、重计算

三、转移+重计算

四、压缩（量化、剪枝以及传统压缩）

→ **解决GPU显存不足问题**

探讨：针对显存优化，AI+Sys未来的研究应该怎么走呢？

Final

Thanks