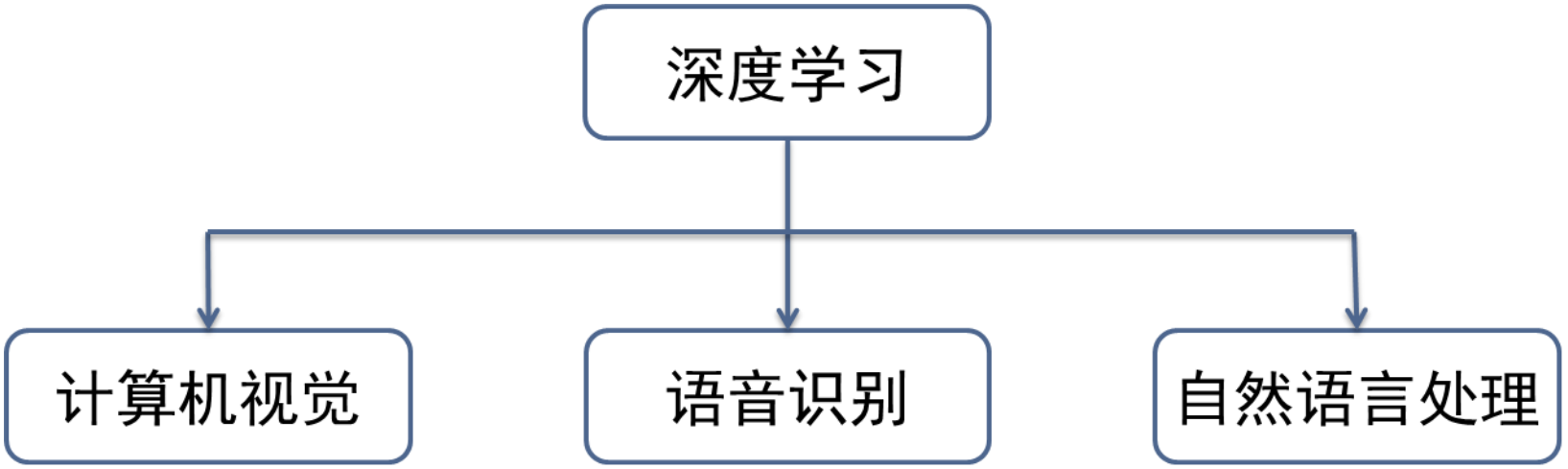
面向超深神经网络训练的动态GPU显存管理系统的设计与实现

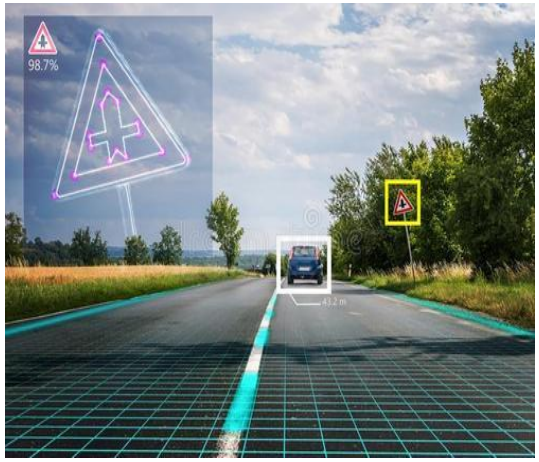
汇 报 人：王彦泽 174115
指导导师：罗军舟教授
汇报时间：06月25日

目录

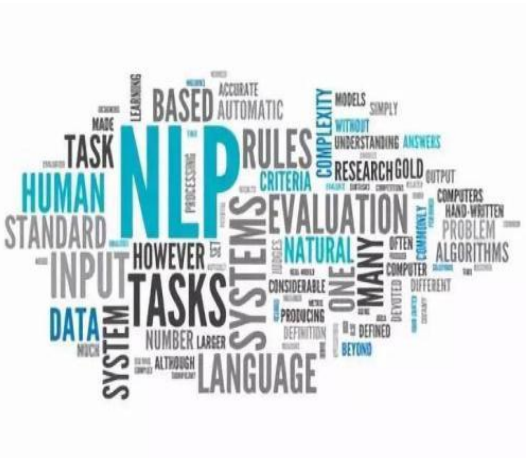
01 课题研究背景

深度学习


深度学习
计算机视觉
语音识别
自然语言处理

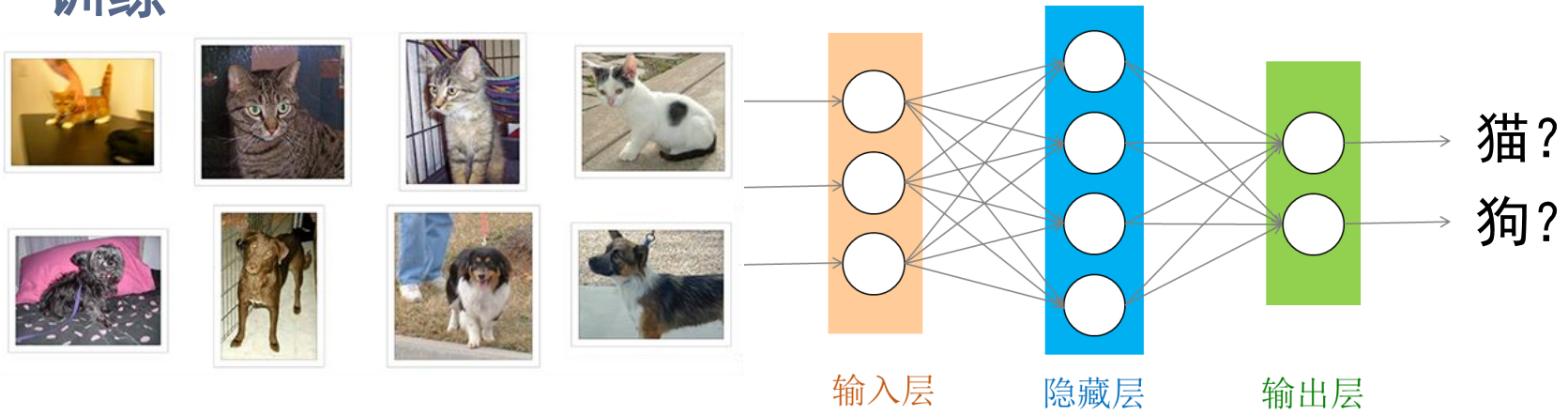

98.7%


YOUR PERSONAL
voice assistant

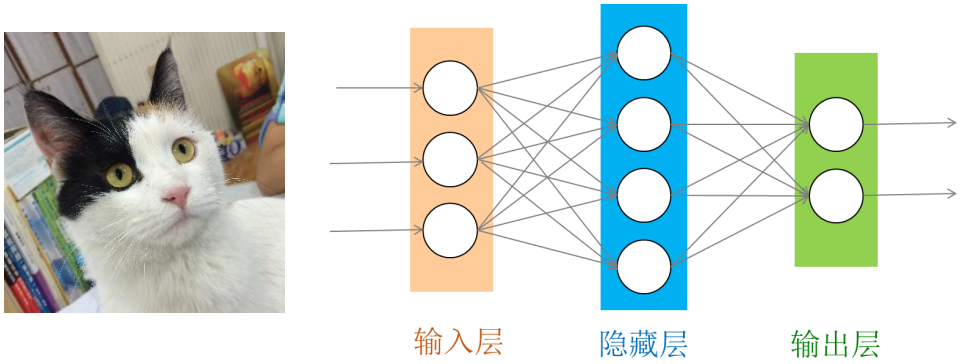

NLP
TASK
BASED
RULES
HUMAN
EVALUATION
STANDARD
INPUT
TASKS
SYSTEMS
DATA
LANGUAGE
RESEARCH
NATURAL
HOWEVER
ALGORITHMS

深度学习

训练



推断

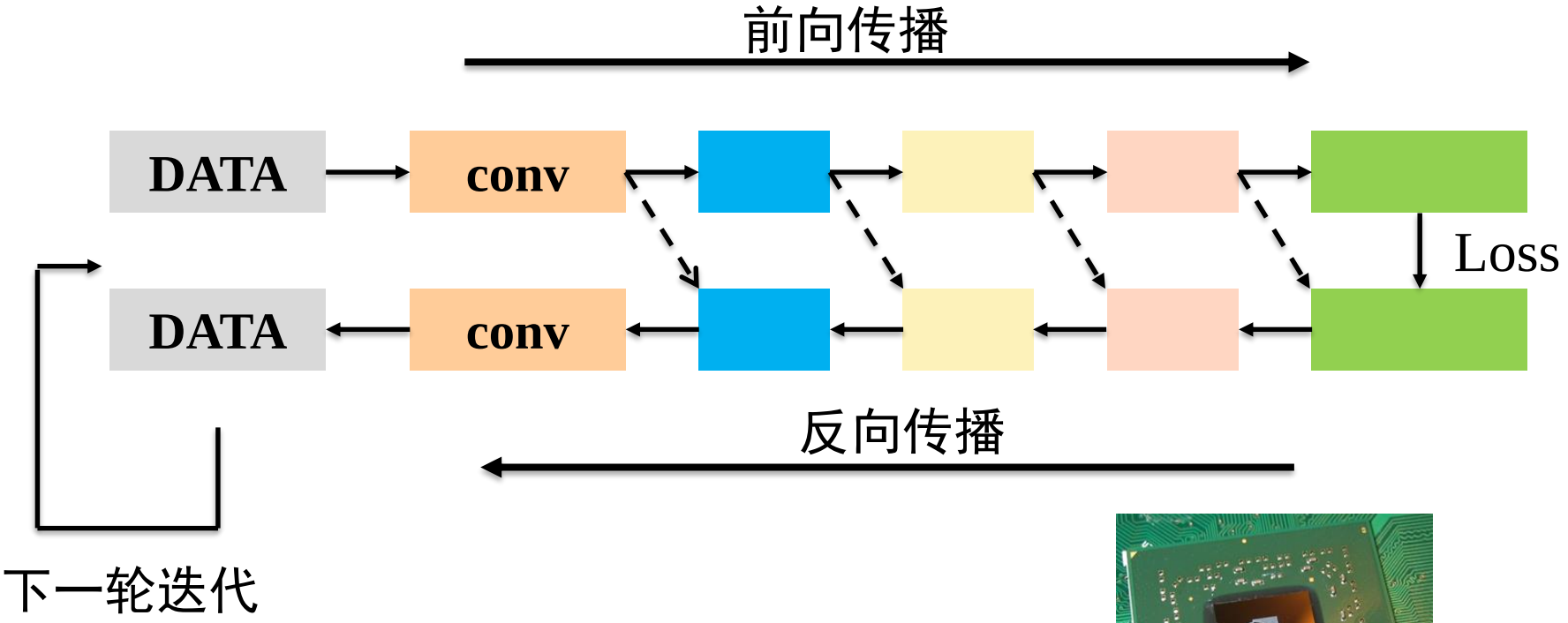


训练时间：推断时间

1×10^{10} ： 1

训练过程

超深神经网络 UDNN
(Ultra-deep neural network)

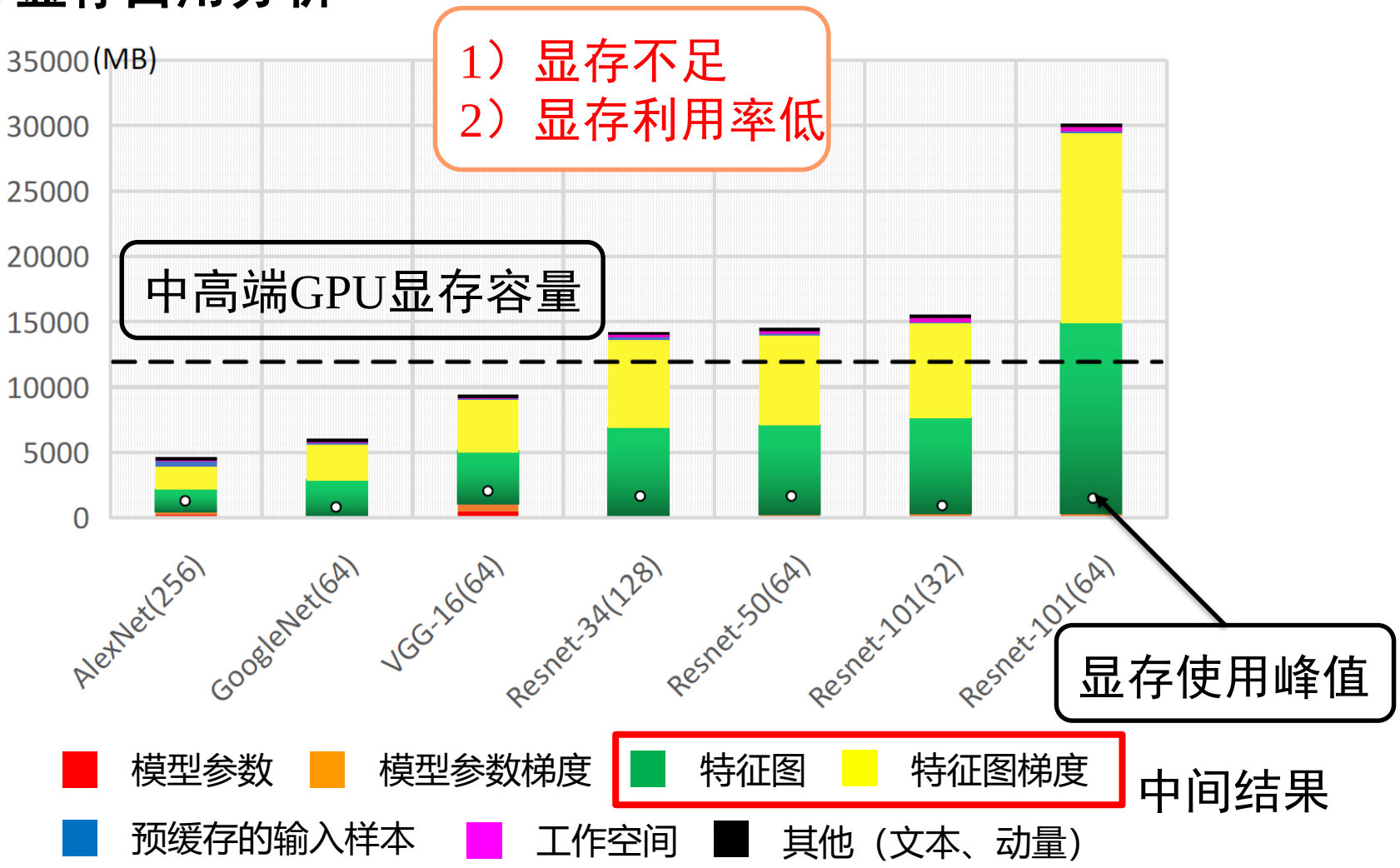


计算和存储密集型任务

GPU

Southeast University

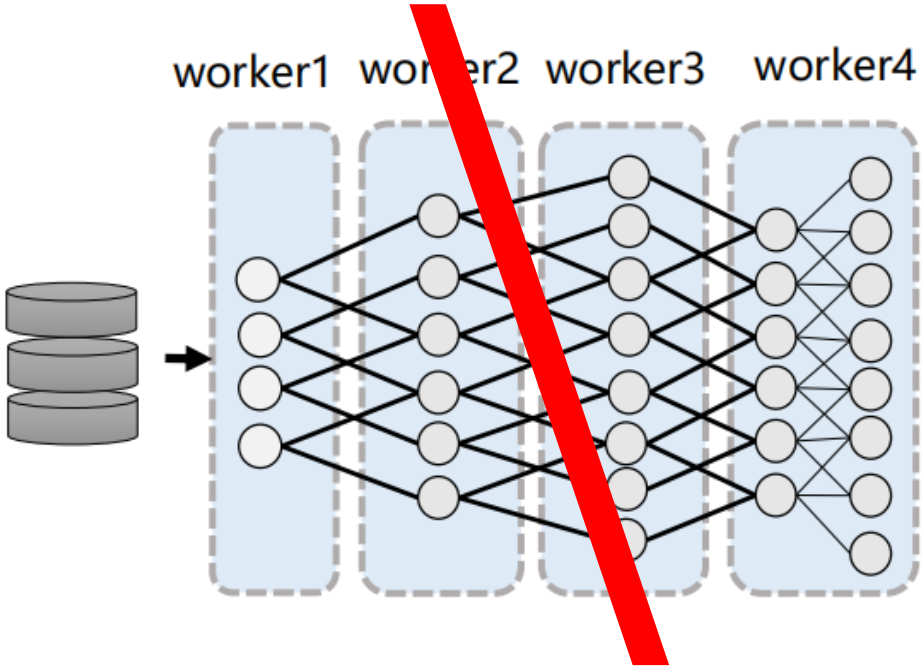
显存占用分析



国内外研究现状

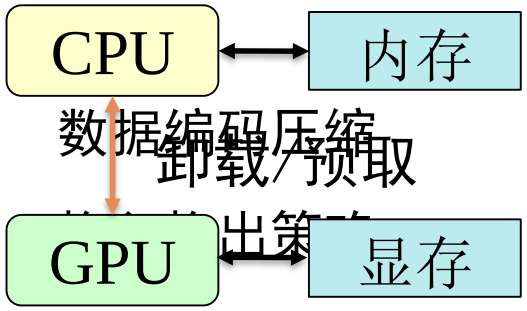
现有的解决GPU显存不足的方法：

1）多GPU模型并行



网络内部传输，通信瓶颈

2）单GPU优化显存使用



传输等待，计算停滞

面临的挑战

1）超深神经网络的可训练性

在显存有限的情况下，如何使UDNN可训练？

2）超深神经网络的训练效率

在优化显存管理的同时，如何保证UDNN的训练效率？

02 研究目标

研究目标

总体目标

优化显存管理，使UDNN在显存有限的情况下可训练，同时提升训练效率。

理论目标

1）通过分析网络结构，计算网络模型中各层与中间结果的数据依赖关系；
2）通过对影响超深神经网络训练效率的资源进行建模量化训练过程并确定限制条件；
3）结合数据依赖关系和限制条件，并根据神经网络训练特征确定显存—内存卸载预取方案。

系统目标

针对训练UDNN时显存不足和优化显存带来的效率降低的问题，确定网络模型中的数据依赖关系，建立训练的性能模型，完善显存—内存卸载预取机制，实现面向超深神经网络训练的动态GPU显存管理系统。

03 研究内容

1）网络结构分析：

本研究将分析给定的超深神经网络模型结构，确定网络中各层的执行顺序，为各层与中间结果建立依赖关系，由依赖关系确定中间结果在训练过程中的存储序列和释放序列。

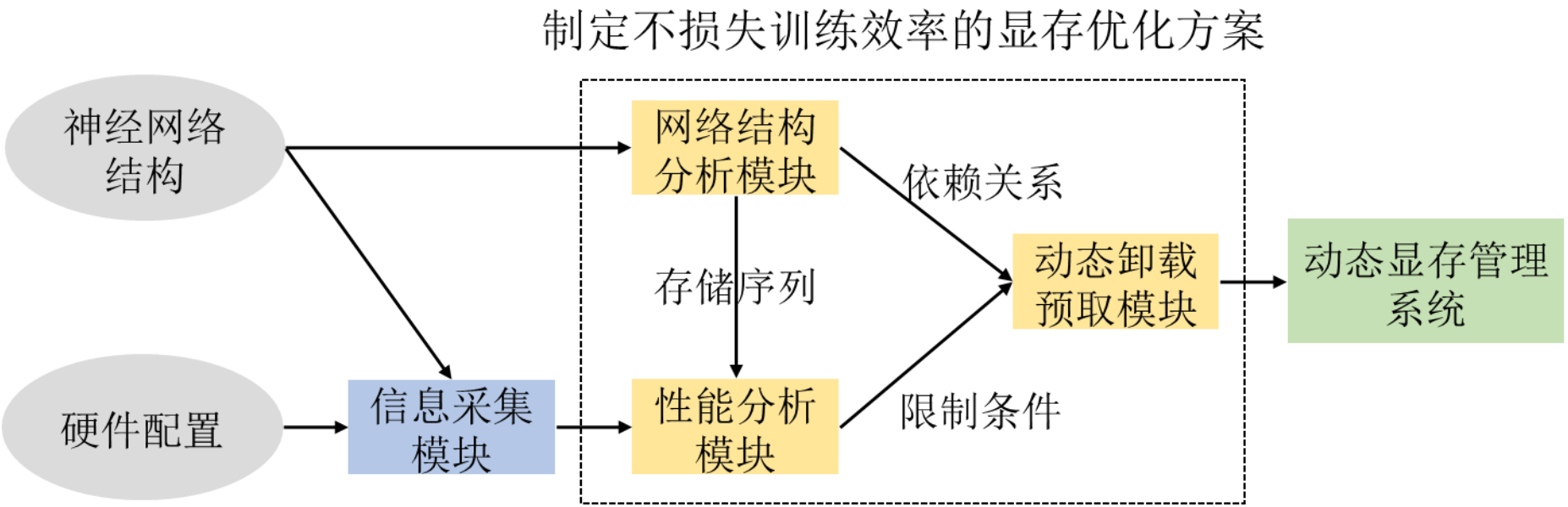
动态显存管理系统

2）性能分析：

根据网络结构和硬件配置，收集建立性能模型所需的信息，综合考虑GPU计算性能，GPU显存使用和PCIe总线的通信性能，对训练过程进行性能建模。

3）优化策略：

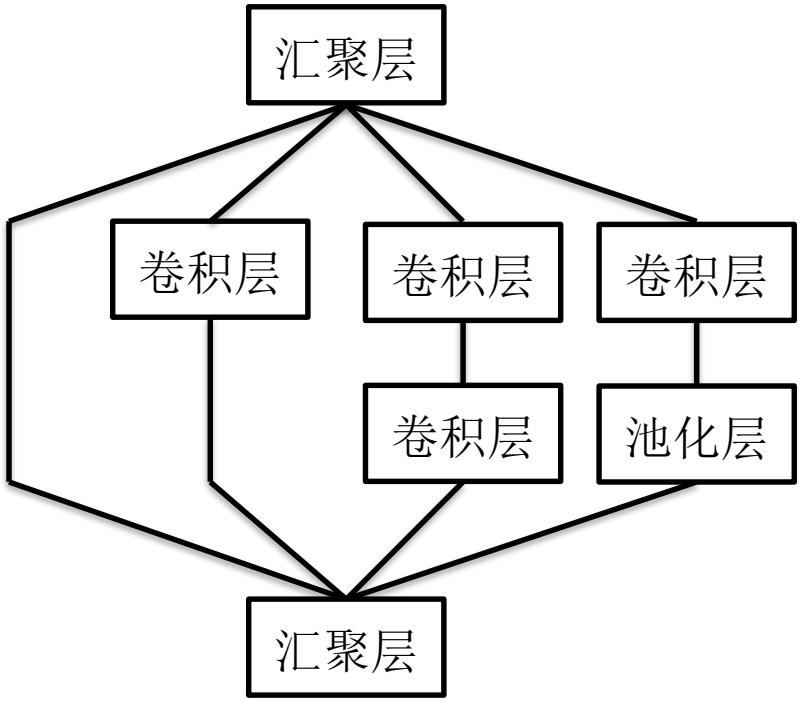
结合性能模型的限制条件和数据依赖关系，并根据神经网络训练的特征，优化显存-内存卸载预取机制。


制定不损失训练效率的显存优化方案
神经网络结构
硬件配置
信息采集模块
网络结构分析模块
性能分析模块
存储序列
依赖关系
限制条件
动态卸载预取模块
动态显存管理系统

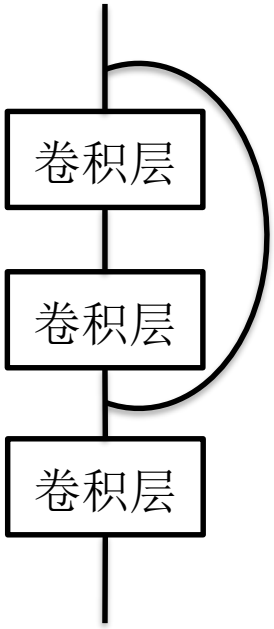
研究内容逻辑关系图

04 实施方案

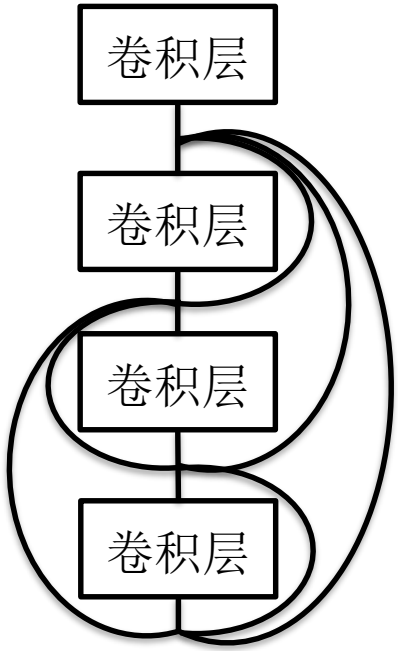
网络结构分析模块



inception v4

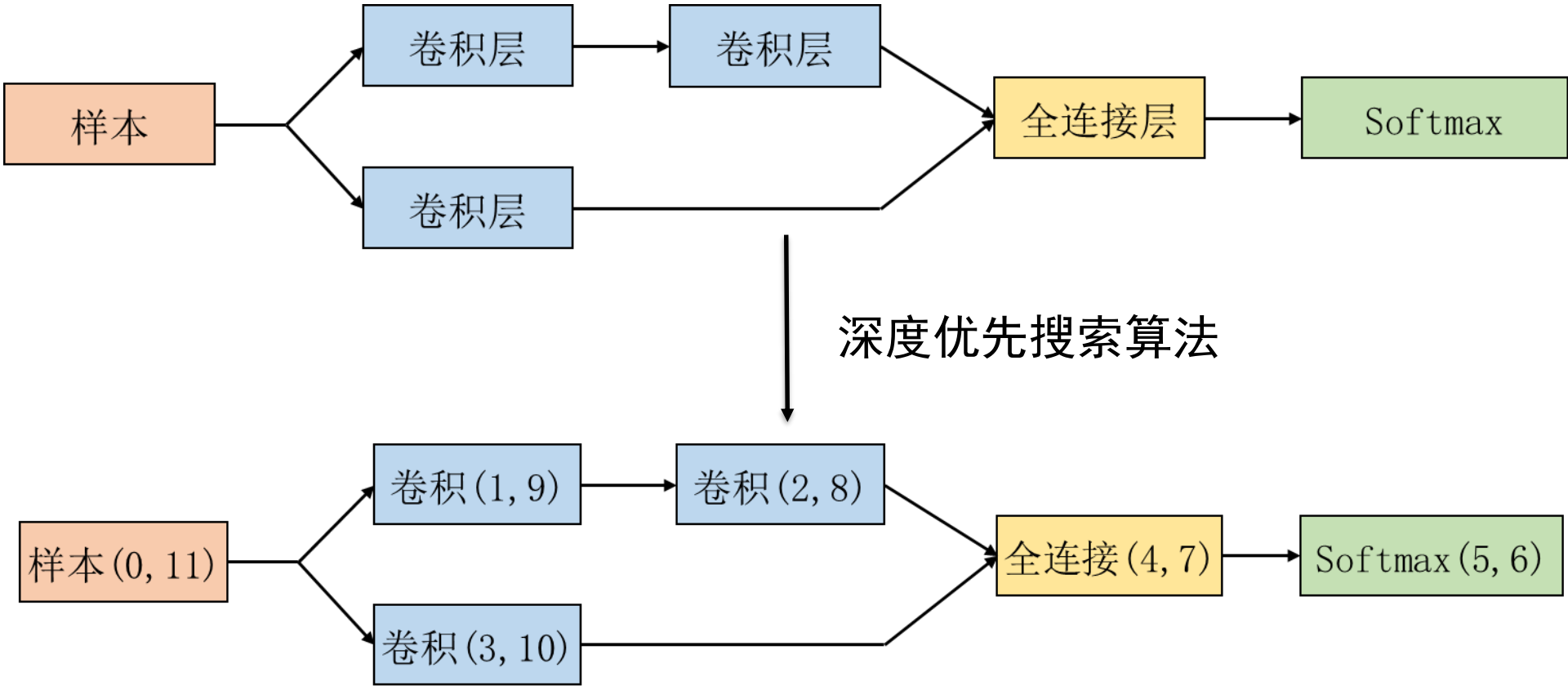


ResNet

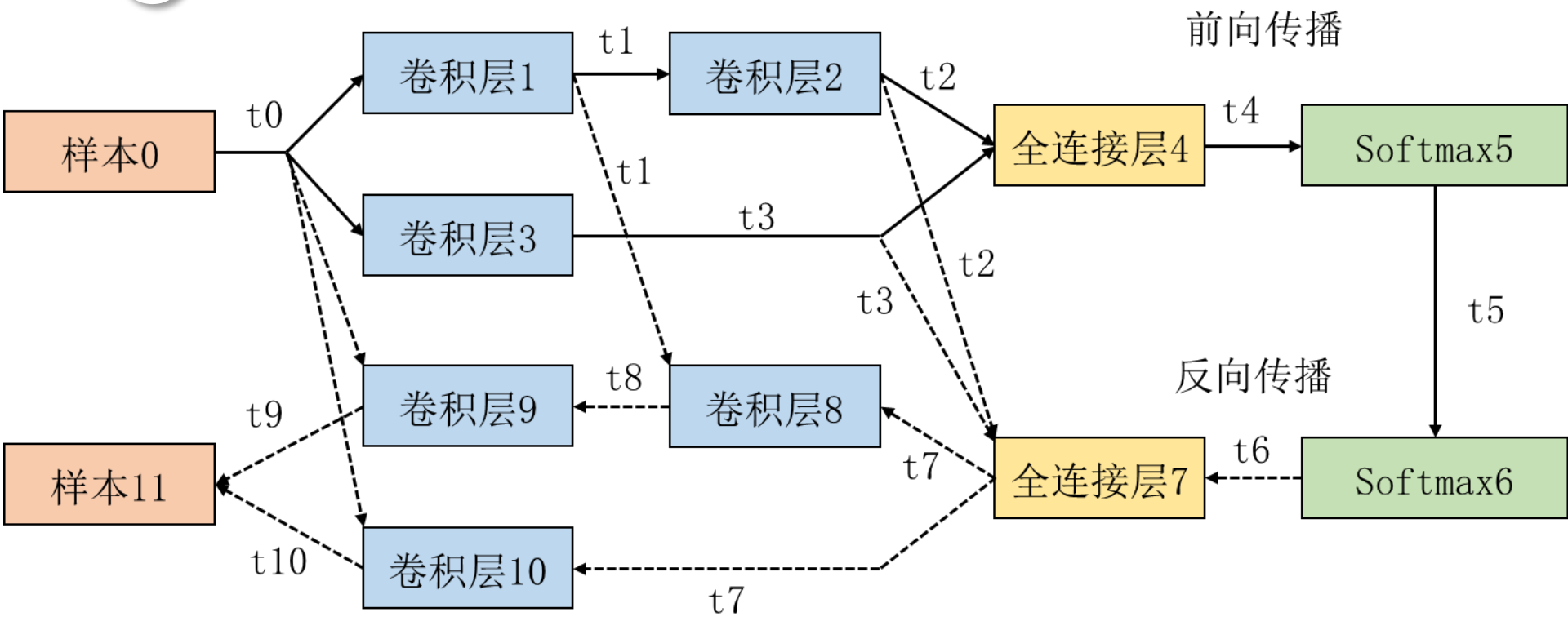


DenseNet

网络结构分析模块

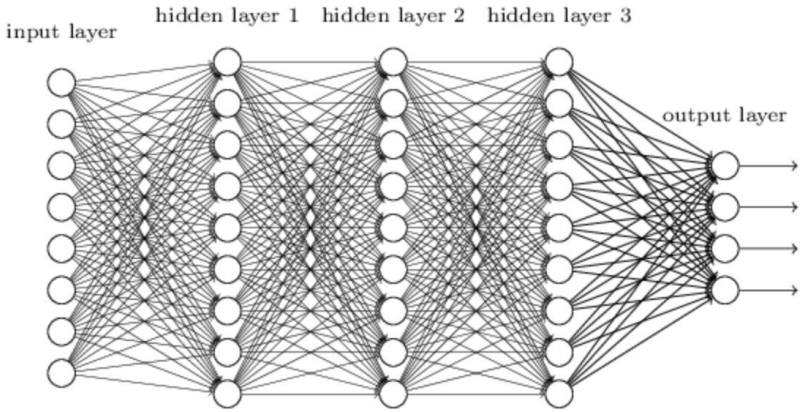

样本
卷积层
卷积层
卷积层
全连接层
Softmax
深度优先搜索算法
样本(0, 11)
卷积(1, 9)
卷积(2, 8)
卷积(3, 10)
全连接(4, 7)
Softmax(5, 6)

网络结构分析模块



Step:1 in:{t0}	Step:2 in:{t1}	Step:3 in:{t0}	Step:4 in:{t2, t3}	Step:5 in:{t4}
Step:10 in:{t0, t7}	Step:9 in:{t0, t8}	Step:8 in:{t1, t7}	Step:7 in:{t2, t3, t6}	Step:6 in:{t5}

信息采集模块



神经网络结构

硬件配置

静态信息：

- 网络模型的基本运算
- 各层中间结果的大小
- 单精度浮点运算量

动态信息：

- 计算时间
- 传输时间

性能分析模块

GPU计算模型

变量	含义
N	UDNN的层数
k	批处理(minibatch)大小
$FLOPs_l$	第l层的单精度浮点运算量
FLOPS	每秒钟GPU处理的单精度浮点运算量
t_l	第l层的计算时间
t_{iter}	一轮迭代的计算时间

$$t_{iter}(k) = \sum_{l=1}^{N} t_l(k) + \sum_{l=N+1}^{2N} t_l(k) = \sum_{l=0}^{2N} \frac{FLOPs_l(k)}{FLOPS(FLOPs_l(k))}$$

性能分析模块

GPU存储模型

变量	含义
$Seq_{allocate}$	当前分配的中间结果个数
$Seq_{realease}$	当前释放的中间结果个数
$M_{allocate[p]}$	第p个中间结果所需的显存容量
$M_{release[q]}$	第q个中间结果所需的显存容量
M	给定的显存预算
$Seq_{allocate}$	当前分配的中间结果个数

$$\sum_{p=0}^{Seq_{allocate}} M_{allocate[p]}(k) - \sum_{q=0}^{Seq_{release}} M_{release[q]}(k) \leq M$$

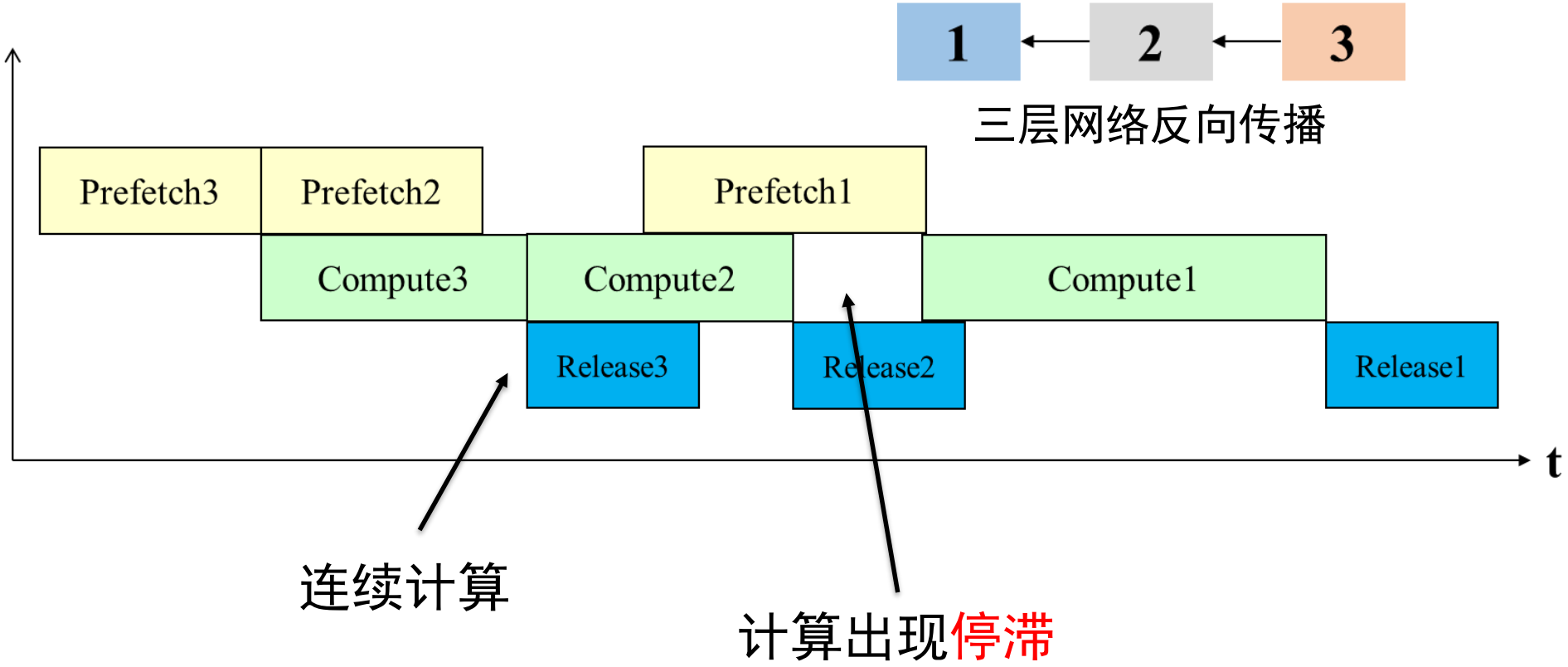
性能分析模块

PCIe通信模型

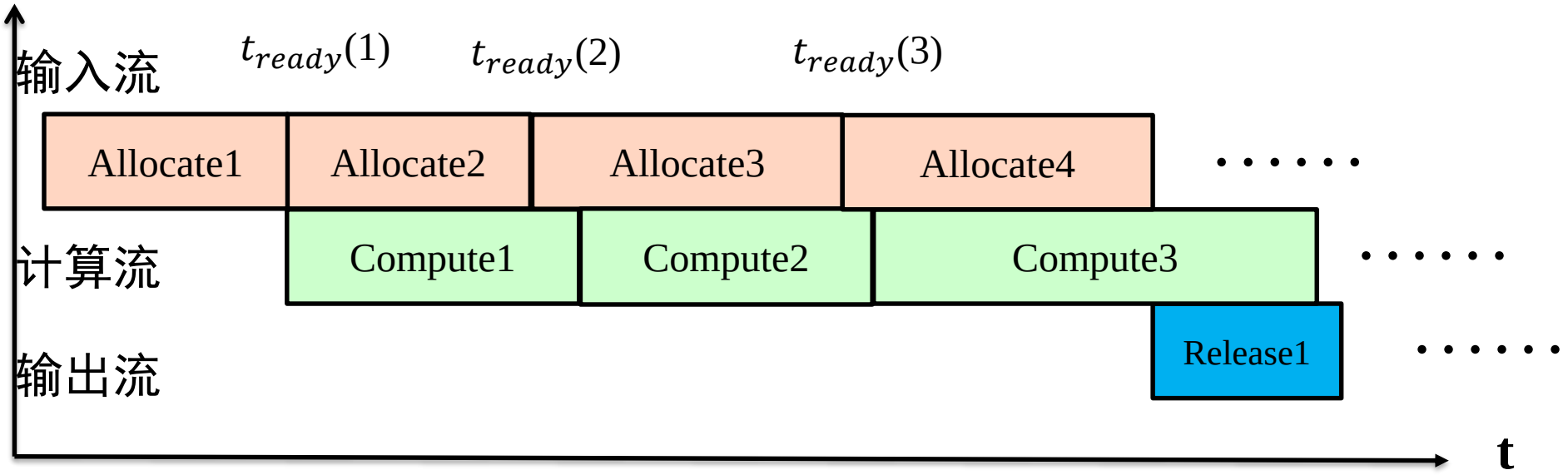
接近实际的PCIe带宽

$$t_{pre/off} = \frac{T_{pre/off}(k)}{Bandwidth_{avail}}$$

动态卸载预取模块


1
2
3
三层网络反向传播
Prefetch3
Prefetch2
Prefetch1
Compute3
Compute2
Compute1
Release3
Release2
Release1
t
连续计算
计算出现停滞

动态卸载预取模块


输入流
$t_{ready}(1)$
$t_{ready}(2)$
$t_{ready}(3)$
Allocate1
Allocate2
Allocate3
Allocate4
计算流
Compute1
Compute2
Compute3
输出流
Release1
t

动态卸载预取模块

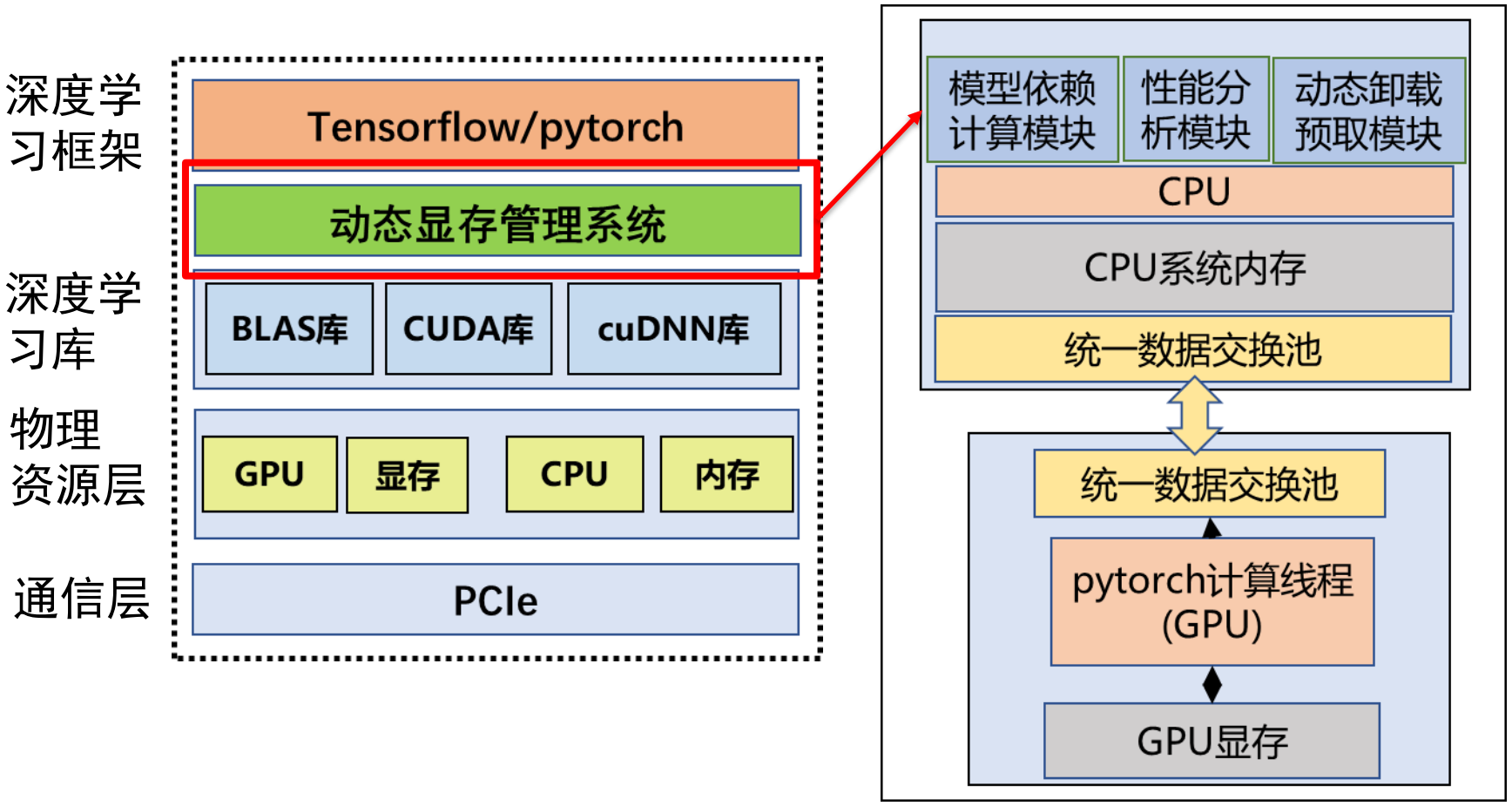
目标表示（在计算前完成显存工作）：

$$t_{ready}(l) \leq \sum_{1}^{l-1} t_{compute}$$

限制条件（显存限制）：

$$M_{used\_present} + M_{allocate[l]} \leq M$$

系统总体设计


深度学习框架
深度学习库
物理资源层
通信层
Tensorflow/pytorch
动态显存管理系统
BLAS库
CUDA库
cuDNN库
GPU
显存
CPU
内存
PCIe
模型依赖计算模块
性能分析模块
动态卸载预取模块
CPU
CPU系统内存
统一数据交换池
统一数据交换池
pytorch计算线程(GPU)
GPU显存

05 进度安排

课题时间安排

起讫日期	工作内容和要求	备注
2018.12—2019.03	研读相关文献，进行相关理论学习	已完成
2019.04—2019.06	整理课题相关的材料， 完成理论分析和方法设计	已完成
2019.07—2020.01	完善理论方案，进行系统开发和测试	
2020.02—2020.04	撰写毕业论文	
2020.05—2020.06	准备毕业答辩	

谢谢！ 敬请老师指正！

THANKS FOR LISTENING