Санкт-Петербургский
государственный университет

Тема: Распределенный инференс глубоких нейросетевых моделей

Выполнил: Колосков Виктор Юрьевич
Научный руководитель: Першин Антон Юрьевич

Постановка задачи

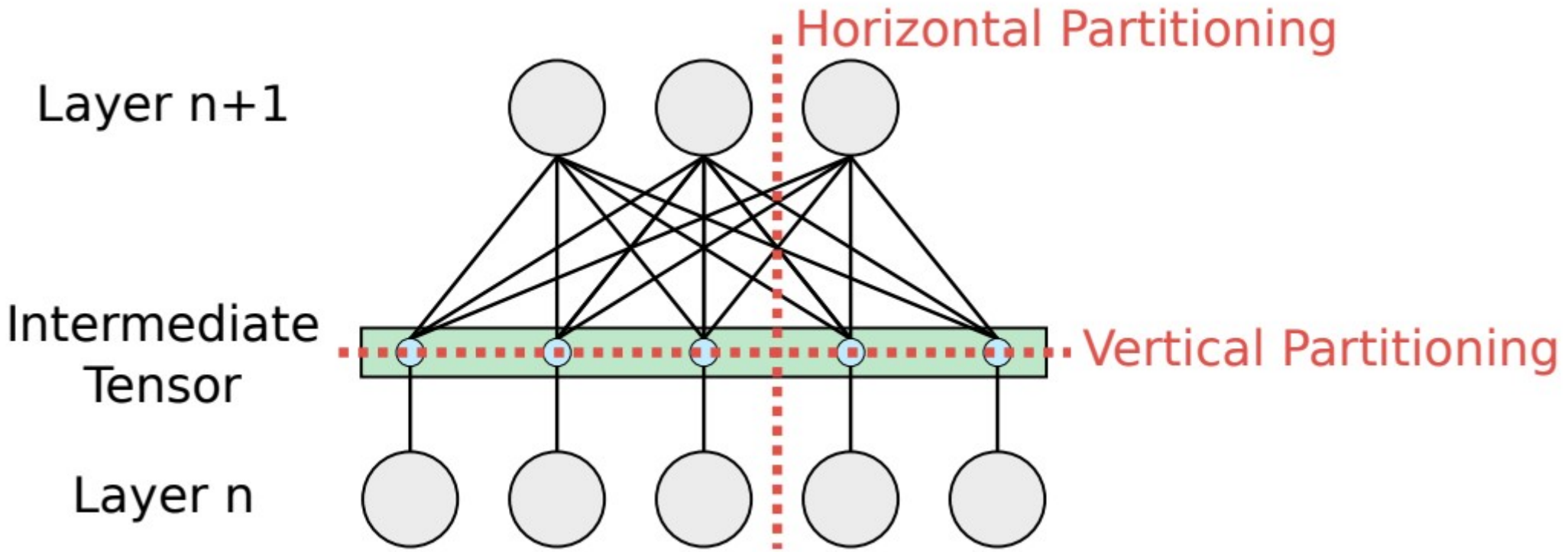
Инференс нейросетевых моделей — это процесс использования уже обученной нейронной сети для предсказания результатов на новых, ранее невидимых данных.

Активное развитие технологий глубоких нейронных сетей привело к необходимости эффективного использования нескольких вычислительных устройств для выполнения инференса. Это обусловлено ограничениями памяти видеокарт и центральных процессоров при работе с крупными моделями.

Для решения этой проблемы возникает необходимость разделить матрицы весов модели между различными устройствами. Такой подход требует создания определенных "разрезов" нейронной сети, которые будут распределены по доступным ресурсам.

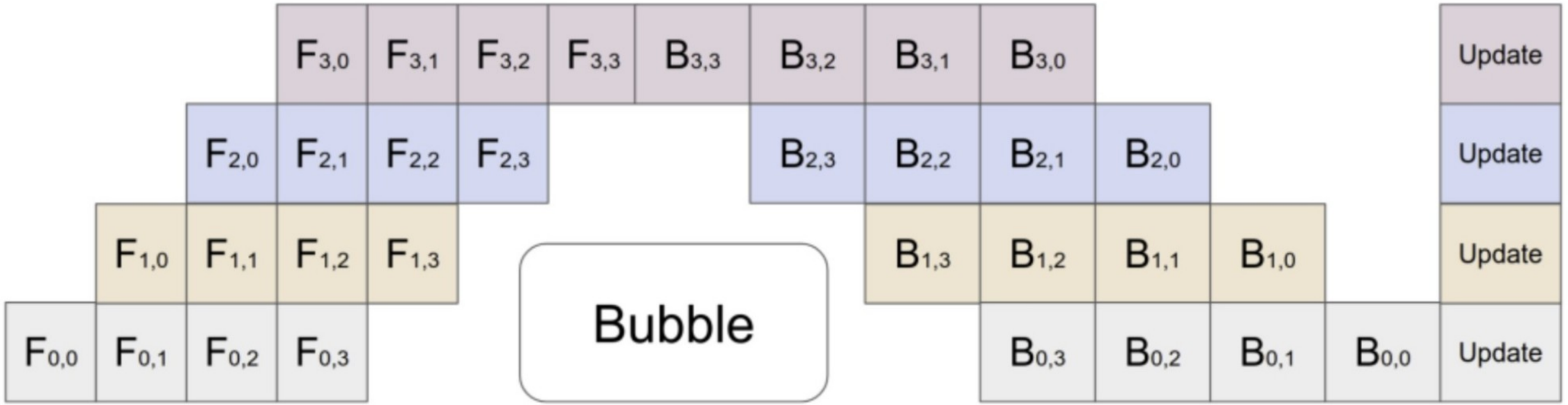
Цель: исследовать различные способы распределенного инференса и инструменты их реализации.

Виды разрезов нейронной сети



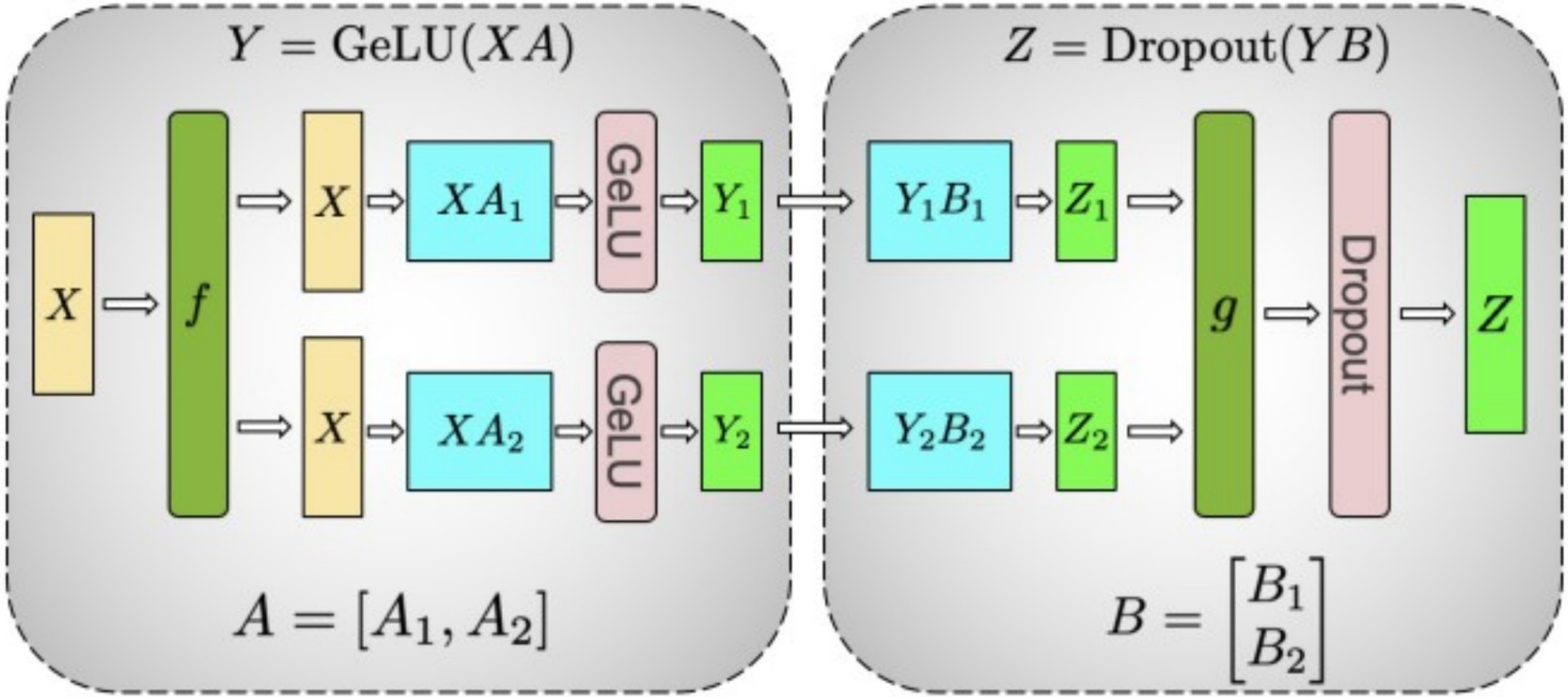
Примеры разрезов нейронной сети

Виды параллелизма моделей: конвейерный параллелизм



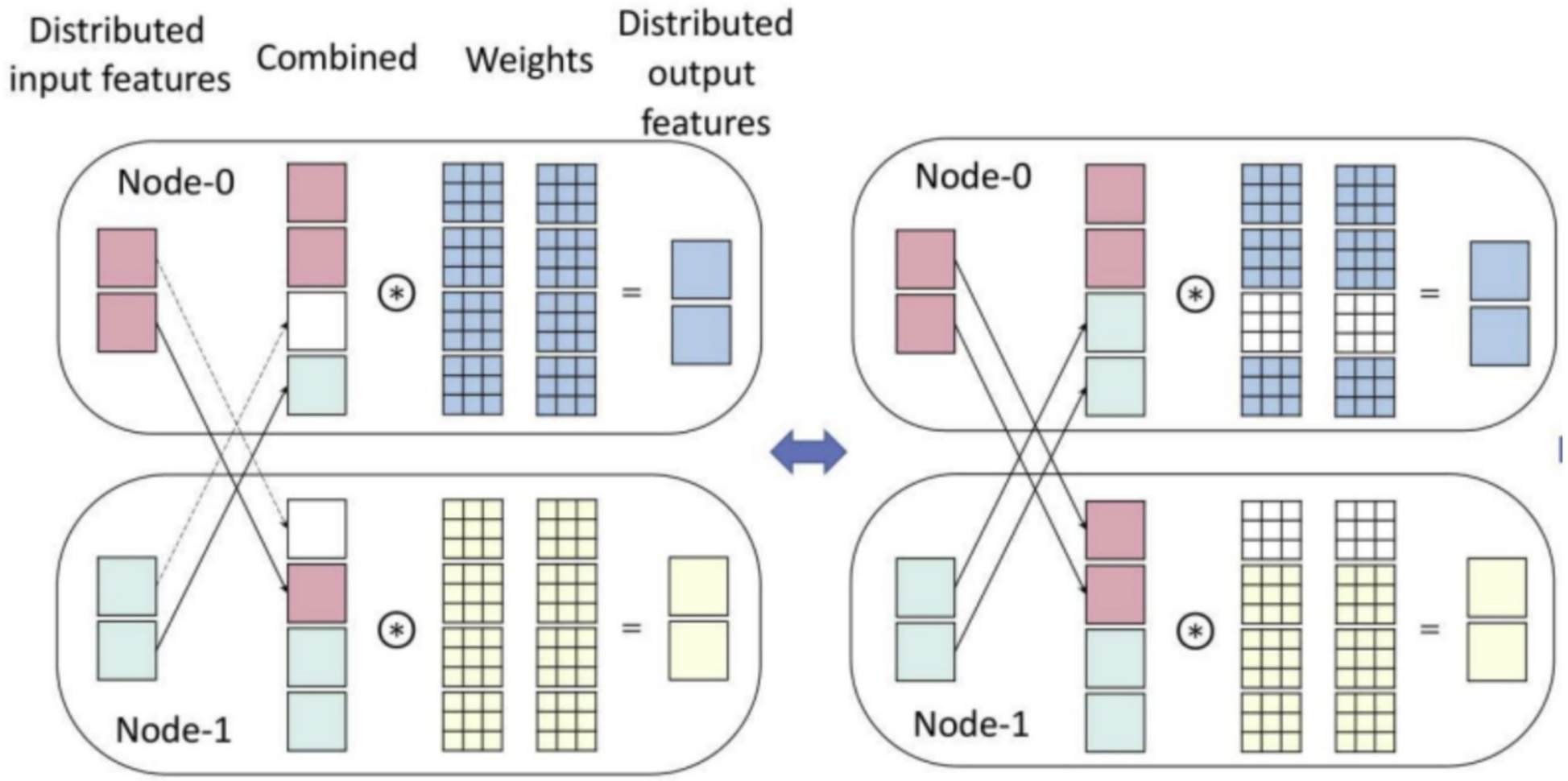
Пример работы планировщика при конвейерном параллелизме

Виды параллелизма моделей: тензорный параллелизм



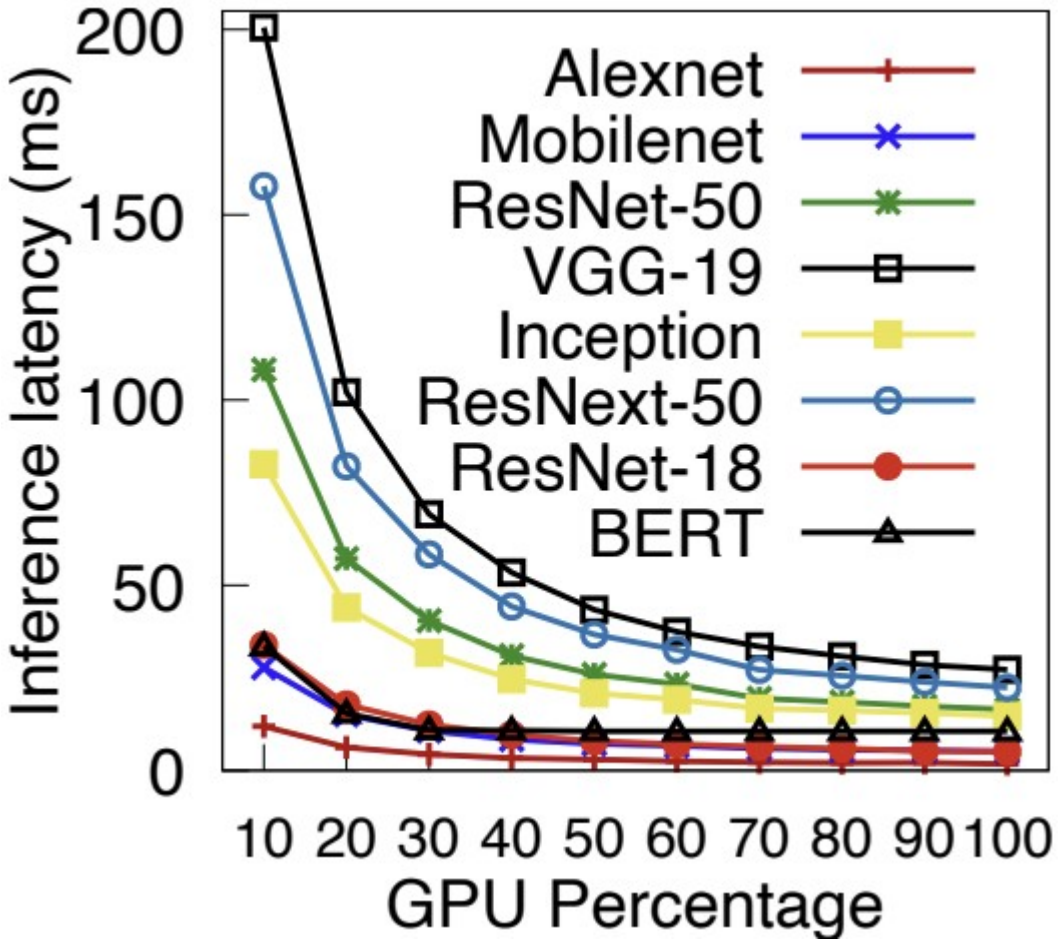
Пример тензорного параллелизма для перцептрона

Методы оптимизации : разреженные коммуникации



Сведение задачи уменьшение коммуникаций между видеокартами к задаче прунинга

Методы оптимизации: мультиплексирование и пространственно-временное планирование GPU



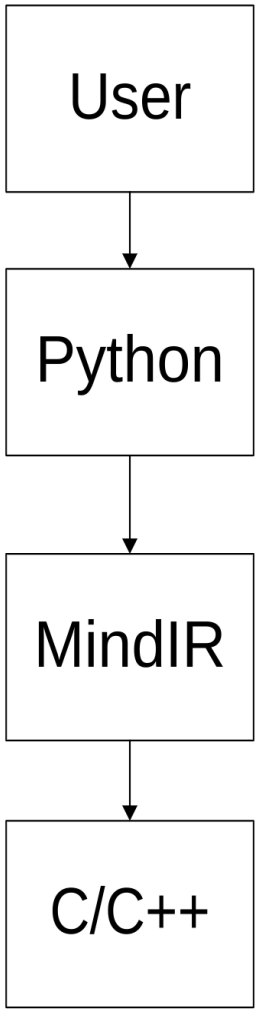
Задержки при инференсе моделей в зависимости от выделенной мощности видеокарты Nvidia V100

MindSpore

MindSpore — платформа глубокого обучения для всех сценариев, направленная на упрощение разработки, эффективное выполнение и унифицированное развертывание для всех сценариев. Фреймворк поддерживает как обучение на одном устройстве, так и распределенное обучение на кластере устройств, включая CPU, GPU и специализированные чипы Ascend. Также MindSpore имеет возможность работать с библиотеками, направленными на оптимизацию вычислений, например, TensorRT

MindSpore: принцип работы

Общий принцип работы



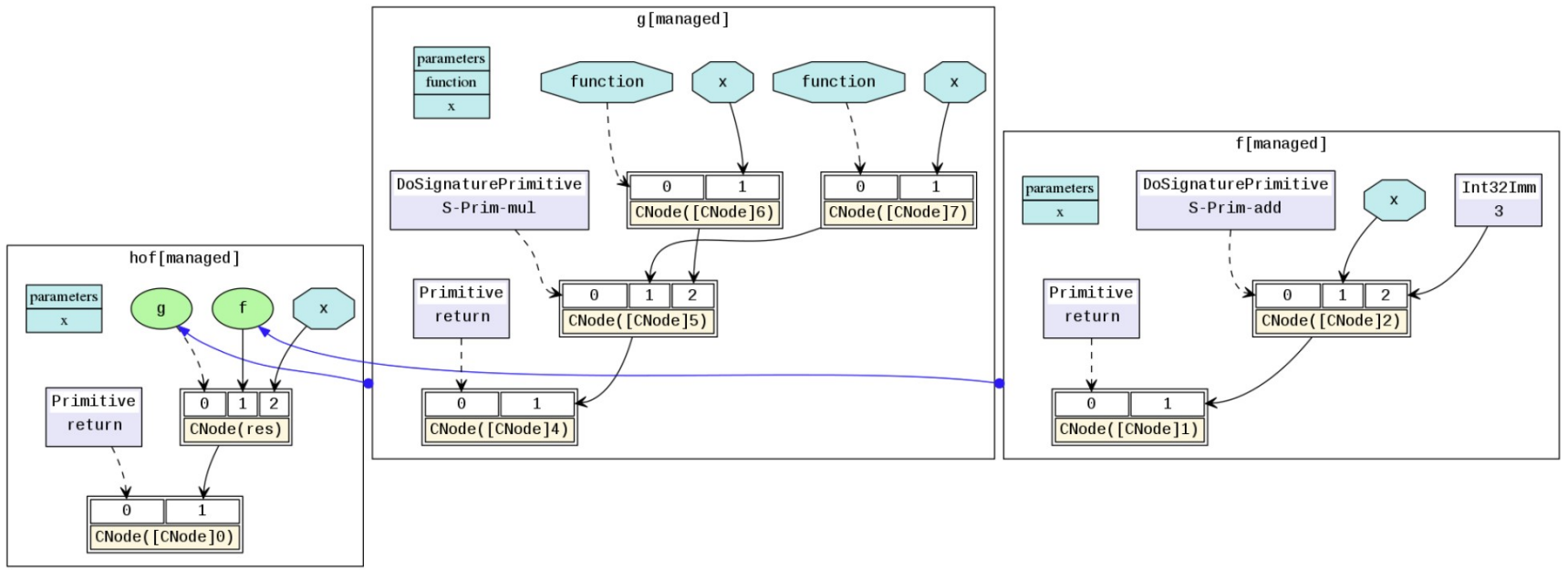
MindIR

```python
import mindspore as ms

def func(x, y):
    return x / y

@ms.jit
def test_f(x, y):
    a = x - 1
    b = a + y
    c = b * func(a, b)
    return c
```

```
lambda (x, y)
    let a = x - 1 in
    let b = a + y in
    let func = lambda (x, y)
        let ret = x / y in
        ret end in
    let %1 = func(a, b) in
    let c = b * %1 in
    c end
```



MindSpore: полуавтоматический параллелизм

1. $Y = X \times W = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \times [\, W_1 \quad W_2] = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$

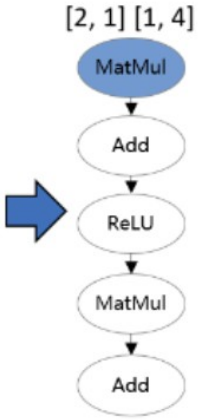
2. $Z = Y \times V = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \times \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}$

	X is sharded in the row dimension W is sharded in the column dimension				**Y is sharded in the row & column dimension V is sharded in the row dimension**				
D1	X_1	$\longrightarrow$ W_1	$\longrightarrow$ X_1W_1	=	Y_{11}	$\longrightarrow$ V_1	$\longrightarrow$ $Y_{11}V_1$	allreduce	Z_1
D2	X_1	$\longrightarrow$ W_2	$\longrightarrow$ X_1W_2	=	Y_{12}	$\longrightarrow$ V_2	$\longrightarrow$ $Y_{12}V_2$		Z_1
D3	X_2	$\longrightarrow$ W_1	$\longrightarrow$ X_2W_1	=	Y_{21}	$\longrightarrow$ V_1	$\longrightarrow$ $Y_{21}V_1$	allreduce	Z_2
D4	X_2	$\longrightarrow$ W_2	$\longrightarrow$ X_2W_2	=	Y_{21}	$\longrightarrow$ V_2	$\longrightarrow$ $Y_{22}V_2$		Z_2

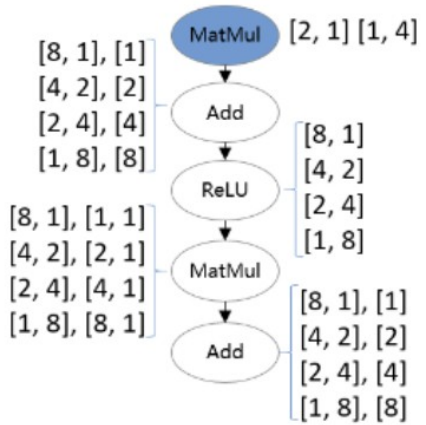
Пример тензорного параллелизма MindSpore

MindSpore: автоматический параллелизм

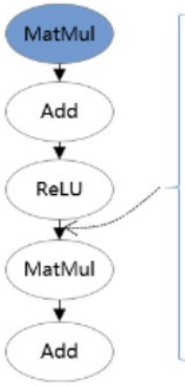
```
class FFN():
def __init__(hidden_size, ffn_hidden_size):
    self.dense = nn.Dense(hidden_size, ffn_hidden_size)
    self.dense.matmul.shard(((2, 1), (1, 4)))
    self.dense2 = nn.Dense(ffn_hidden_size, hidden_size)
    self.relu = P.ReLU()
def construct(x)
    x = self.dense(x)
    x = self.relu(x)
    x = self.dense(x)
    return x
```



(a) Transform from a model defining script to a computation graph

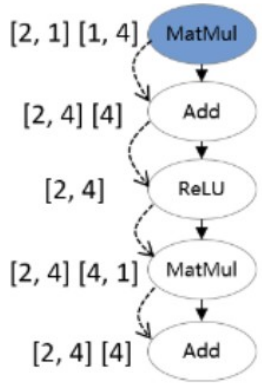


(b) Enumerate strategies for non-configured ops



s_strategy	t_strategy	cost
...	...	...
[2, 4]	[8, 1], [1, 1]	AllGather
[2, 4]	[4, 2], [2, 1]	AllToAll
[2, 4]	[2, 4], [4, 1]	0
[2, 4]	[1, 8], [8, 1]	AllGather
...	...	...

(c) Enumerate strategies and costs for Tensor Redistribution on each edge. Due to the space limit, only a part of strategies on edge ReLU->MatMul is listed



(d) Propagate from the configured op to other ops

Алгоритм поиска стратегии распространения осколков

MindSpore: практические эксперименты

Для экспериментов используется платформа с двумя графическими ускорителями. В качестве модели взят трехслойный перцептрон. Первый слой состоит из матрицы весов, размера 784x512 и операции ReLU. Второй слой представлен в виде матрицы 512x512 и ReLU. Третий слой состоит из матрицы размера 512x10. В качестве датасета для обучения использован MNIST. Количество эпох - 10, на каждой эпохе производится 1875 шагов. Тензорный параллелизм может сильно зависеть от стратегии разбиения, поэтому в эксперименте берется разбиение, полученное при использовании алгоритма поиска стратегии распространения осколков. При конвейерном параллелизме на первом GPU вычисляются данные первого слоя, на втором - второго и третьего слоя соответственно. В таблице ниже представлено время, затраченное не обучение модели в зависимости от типа используемого параллелизма.

Тип параллелизма	Время (мин)
Конвейерный параллелизм	2.81
Тензорный параллелизм	2.57
Без параллелизма	3.90

Результаты экспериментов