Inter-layer Scheduling Space Size Calculation for SET

This document outlines the process of calculating the size of the inter-layer scheduling space defined in the SET paper. We will simplify the problem, and then show the way to calculate the asymptotic size of the scheduling space, $O((5+2\sqrt{6})^n)$ (where n is the number of layers).

In the following paragraphs, denote n as the number of layers in the network.

Problem Specification

The RA Tree is a notation introduced in the SET paper. It is a tree which has the following properties:

(1) There are n **leaf nodes** in the tree, each corresponds to a layer in the DNN.

(2) Each **inner node** can choose one of **two types** (S or T).

(3) The **leaves** are ordered in a **topological order** under the DAG dependency of the layers in the DNN.

(4) Each node has a **"batch size"**, and the batch size of a child is a factor of the batch size of the parent.

As stated in the paper, we only consider the structure of the tree, which means we only consider property (1) and (2) in the calculation.

This is because for (3) we only need to choose one of the possible topological orders in the DAG, which means we only need to multiply the space size with the number of topological orders in the DAG. This number differs with the DAG topology and is hard to estimate generally. For (4) we need to count the factors of specific batch sizes, which is very hard since integer factoring is a hard problem. Also batch size is less essential in the RA Tree. *Considering (3) and (4) will further enlarge the calculated space size.*

In conclusion, we will calculate the number of RA Trees under property (1) and (2), which is **the number of n-leaf trees where each inner node has one of two types**.

Space Size Calculation

Firstly, if we ignore the type of leaf nodes, then the problem is simplified into "counting the number of n-leaf-node trees". This number is called *the Small Schroeder Numbers* (A001003 in OEIS), and is well studied.

Now, if we add d types for each inner node (in our case $d = 2$), this becomes *the Weighted Small Schroeder Numbers* (A107841). Although less famous, it is also studied by previous works. This work [1] defines the Weighted Small Schroeder Numbers as

$$s_d(n) = \sum_{k=0}^{n-1} s(n,k)d^k$$

where $s(n,k)$ is the number of trees with n leaf and k inner node. By multiplying d^k, this is the number of such trees where each inner node has d types to choose. Then by summing over k, $s_d(n)$ will be the number of n-leaf trees with d inner node types, and the size of the space we want to calculate is $s_2(n)$.

The work has the following proposition (proposition 4 in the work [1]):

Proposition: For $d > 0$, as $n \to +\infty$,

$$s_d(n) \sim \left(\frac{(\sqrt{d+1} - \sqrt{d}) \cdot d^{1/4}}{2(d+1)^{3/4}\pi^{1/2}} \right) \cdot n^{-3/2} \cdot \left(2d + 1 + 2\sqrt{d^2 + d}\right)^n$$

Notice that, the first term in the right-hand side is a constant, so in the language of big-theta notation, this is equivalent to:

$$s_d(n) = \Theta\left(n^{-\frac{3}{2}} \cdot \left(2d + 1 + 2\sqrt{d^2 + d}\right)^n\right)$$

Since power functions ($n^{-\frac{3}{2}}$) grows much slower than exponential functions ($\left(2d + 1 + 2\sqrt{d^2 + d}\right)^n$), we can remove the power part and use big-o notation:

$$s_d(n) = O\left(\left(2d + 1 + 2\sqrt{d^2 + d}\right)^n\right)$$

Then by substituting $d = 2$ we will get

$$s_2(n) = O\left(\left(5 + 2\sqrt{6}\right)^n\right)$$

which is a lower-bound estimation of our space size, since considering property (3) and (4) will bring much more possibilities.

References

[1] Yu Hin Gary Au. Some properties and combinatorial implications of weighted small schröder numbers. *Journal of Integer Sequences*, 24(2):3, 2021.