Machine Learning for Bin Packing Problem

CS3308 Machine Learning

November 28, 2024

1 Introduction

- Three students at most form a group.
- Deadline for this project is January 5, 2025, 23:59. Each group needs to submit a report pdf on Canvas and the source code is also required by providing the link to your github repo.
- This project will be evaluated from workload (20%), model performance (20%), results analysis (40%) and report writing (20%). It is more crucial to conduct a reasonable analysis of the experimental results.
- The bin packing problem (BPP) is a classic optimization conundrum in mathematics and computer science. It involves efficiently packing objects of different sizes into containers entirely and without overlap. The goal is to pack as many items as possible into containers of fixed size, minimizing wasted space.

2 Task1: Policy learning for BPP problem

In this task, you need to first generate training data and testing data by yourself, and then train the policy network for the BPP problem using either supervised learning or reinforcement learning. Due to the variable number of items to be placed, an encoder-decoder framework is often employed to treat the BPP policy as an sequence-to-sequence translation problem. You can refer to such designs to implement your own architecture for solving the BPP problem. More related details can be found in the reference paper.

Hint: Assume the size of container is $100 \times 100 \times 100$, and the generation process is presented in Algorithm 1.

3 Task2: Neural guided search algorithm

Search algorithms are commonly used to solve the BPP problem. In this task, you need to combine the policy network obtained from Task 1 with tree search algorithms (such as depth-first search, breadth-first search, beam search, A* search, Monte Carlo tree search and so on) to address the BPP problem.

Algorithm 1: Bin Packing Problem Generator

Initialize the items list $I \leftarrow \{(100; 100; 100)\}$.
Sampel N from $[10, 50]$.
while $|\mathcal{I}| < N$ **do**
  Pop an item randomly from $\mathcal{I}$ by the item's volume.
  Choose an axis randomly by the length of edge.
  Choose a position randomly on the axis by the distance to the center of edge.
  Split the item into two items.
  To randomly rotate the newly generated objects.
  Add them into $\mathcal{I}$. The relative position in the container and the rotation can be saved as the label for training.
end while
return $\mathcal{I}$

4 Task3: E-commerce packaging problem

The data consists of multiple orders (sta_code), with each order comprising several items (sku_code). Each item is characterized by four attributes: length, width, height, and quantity (qty). The test data is provieded in the attachment. You need to design an algorithm to efficiently utilize multiple courier packaging boxes to pack all items within an order. In the first two tasks, only one container was used, while the third task allows for the use of multiple containers. You can solve this task either by training neural networks as in tasks 1 and 2, or any other strategies designed by yourself with the help of algorithms learned in the class.

Hint: The result is measured by the ratio of the sum of the volumes of all packed items and the sum of the volumes of all containers used.

The available container sizes are as follows: $(35, 23, 13)$, $(37, 26, 13)$, $(38, 26, 13)$, $(40, 28, 16)$, $(42, 30, 18)$, $(42, 30, 40)$, $(52, 40, 17)$, $(54, 45, 36)$.

You need to specify in which container each item should be placed, the position within the container where the item should be placed, and whether the item needs to be rotated.

5 Project Report

Each group is required to turn in a project report with your main ideas, utilized methods and algorithms, experimental settings, finally experimental results, and your discussion about the results. The project report (.pdf) can be written either in English (encouraged) or in Chinese.

At the end of the report, please attach the contribution of each member as a percentage. And work done by each student is needed to be clarified. For example,

Name	Student ID	Score	Work
A	00000000000	30%	-
B	00000000001	30%	-
C	00000000002	40%	-

You are also required to submit the source code of your classification model by providing the link to your github repo in the report. If you do not know how to use github, please visit its tutorial (`https://guides.github.com/activities/hello-world/`) for some advice

6 Reference material

Que, Quanqing, Fang Yang, and Defu Zhang. "Solving 3D packing problem using Transformer network and reinforcement learning." Expert Systems with Applications 214 (2023): 119153.

Laterre, Alexandre, et al. "Ranked reward: Enabling self-play reinforcement learning for combinatorial optimization." arXiv preprint arXiv:1807.01672 (2018).

Zhang, Jingwei, Bin Zi, and Xiaoyu Ge. "Attend2Pack: Bin packing through deep reinforcement learning with attention." arXiv preprint arXiv:2107.04333 (2021).

Zhu, Qianwen, et al. "Learning to pack: A data-driven tree search algorithm for large-scale 3D bin packing problem." Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021.

Pejic, Igor, and Daan van den Berg. "Monte Carlo tree search on perfect rectangle packing problem instances." Proceedings of the 2020 genetic and evolutionary computation conference companion. 2020.