Hyperparameter tuning with search space partition: investigating LA-MCTS

Qianyi Liu
Hyperparameter tuning of large-scale deep learning models
  ○ computationally expensive
  ○ high-dimensional search space

Want fast and efficient hyper-parameter tuning algorithms

A black-box optimisation problem

Bayesian Optimisation / Evolutionary search methods: efficient in lower-dimensional spaces

Partition the hyperparameter space into smaller subspaces

**LA-MCTS**: learn to partition the search space using MCTS + Bayesian Optimisation on promising regions
LA-MCTS

- Learns to partition the search space for Bayesian Optimisation
- Evaluated against baselines on RL tasks and synthetic functions
Goals of this project

- Investigate how LA-MCTS can be applied to tuning hyperparameters of deep neural networks (e.g. ResNet)
  - Mentioned that can be applied to hyper-parameter optimisation
  - Few examples online for this use case
  - Interesting to investigate how this can be done
- Explore how LA-MCTS can be expanded to support parallel tuning
  - Necessary for computational intensive tasks like hyperparameter tuning
  - High cost of doing sampling and evaluation
Parallelism approaches for MCTS

- Leaf parallelization
- Root parallelization
- Tree parallelization with global mutex
- Tree parallelization with local mutexes

○ = Tree node
↓ = Thread location
□ = Locked memory section
‡ = Tree-independent simulation
Directions

- Leaf level parallelisation
  - GPyOpt: parallel Bayesian optimization
  - Extend to distributed tuning?

- Root parallelisation
  - Start multiple LA-MCTS processes, combine the results in the end
  - Since there is no persistent tree structure, essentially just combining the set of samples
  - Ray Tune
Workplan

● Preliminary research
  ○ Read the relevant papers on LA-MCTS ✓
  ○ Get familiar with the codebase of LA-MCTS ✓
  ○ Identify potential places and approaches for parallelism ✓❓

● Implementation + Evaluation
  ○ Application in hyperparameter tuning
    ➢ Tune hyperparameters of ResNet, compare with vanilla BO baseline
  ○ Parallel-optimisation extensions
    ➢ Implement the two methods
    ➢ Evaluate how they scale

● Write Up
Questions