Learning Dynamic Networks
Kale-ab Tessera, Chiratidzo Matowe, Arnu Pretorius, Benjamin Rosman, Sara Hooker

Problem Statement
Overparameterized models have led to many breakthroughs in machine learning.

Challenges:
- Larger Models:
  - Efficient storage and inference.
  - Efficient training of large models.
- Overfitting, regularization and generalization.
- Train Longer:
  - Handle temporal dynamics of training.

Train Longer - Schedules
When we train longer -> more critical temporal decisions to make.

Temporal Decisions (examples include):
- Learning Rate - Initial Learning Rate & LR Schedule.
- Sparsity - Initial Sparsity & Sparsity Schedule.

What about these choices per layer?! - Layerwise Schedules.

Standard approach - choose these schedules through trial-and-error.

Question
Can we learn temporal (possibly layerwise) schedules in a principled manner?

Related Work
- Constant - SET [1], Deep Rewiring (DeepR) [2] and Neural Network Synthesis Tool (NEST) [3].
- Cosine - RigL [4] and Sparse Network From Scratch (SNFS) [5].
- Cubic - [6],[7].

Examples of Simple Schedules:

Overparameterization - Sparsity/Pruning
Common method to handle challenges of overparameterization - Pruning.

Benefits - similar performance, with a fraction of the weights, faster training and more robust to noise.

Our Approach - Can we learn these schedules using RL

Results

1. Learned Schedules are Layerwise Diverse

Conclusion:
In this work, we demonstrate that it is possible to learn well performing dynamic sparsity schedules using reinforcement learning. The schedules learned are not arbitrary and are distinct per layer and pruning method.

References


ResNet-18

Table 1: Test Accuracy (mean and standard deviation) of different schedules on CIFAR-10, using Simple-CNN.

- Non-stationary environment.
- Worse for challenging networks - use techniques like data augmentation and learning rate decay (e.g. ResNet-18).
- High dimension action and (possibly) state space.
- Slow convergence: 25-50 episodes.