## AMC: AutoML for Model Compression and Acceleration on Mobile Devices

**Authors: Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Lil, and Song Han**

# Introduction

There is a need for model compression so as to deploy neural network models on mobile devices which have limited computation resources. In this work, the authors propose AutoML for Model Compression (AMC), with which they achieve state-of-the-art model compression results, in a fully automated way. In deeper networks, the layers are correlated in a non-trivial way, and it is infeasible to perform manual model compression and achieve sub-optimal results. Thus, the authors propose AMC, which leverages reinforcement learning which samples a design space, thus improving the model compression quality.

Searching over a discrete space is disadvantageous, as the compressed modelis very sensitive to the sparsity of each layer. Thus, the authors come up with a compression ratio control strategy. This is done so through trials and errors, penalizing error loss while encouraging model shrinking and speedup. After the policy search is done, the best-explored model is fine-tuned to achieve the best performance. The authors also propose a resource-constrained compression to achieve the best accuracy for latency-critical applications. This is achieved by constraining the search space such that the model compressed is always below the resources budget. The authors demonstrate the applicability of this proposed AMC method by evaluating it on VGG, ResNet, MobileNet-V1.

# Methodology

One of the major goals of this work is to automatically find the redundancy for each layer characterized by sparsity for each layer. A reinforcement learning agent is trained to predict the action and give the sparsity. The agent is encouraged to be smaller, faster and spit out more accurate models. For fine-grained pruning, the sparsity is defined as the number of zero elements divided by the number of total elements. AMC leverages reinforcement learning for efficient search over the action space. Most of the existing works use discrete space as a coarse-grained action space. Though they are not a problem for higher-layer architectures, in the case of model compression, they are very sensitive to the sparsity ratio. Thus, a continuous action space is preferred by the authors for this work.

The reinforcement learning agent receives an embedding state of a particular layer from the environment, from which, it outputs a sparsity ratio as an action. The authors use deep deterministic policy gradient for continuous control of the compression ratio, the noise for which is initialized as 0.5 and decayed after every episode exponentially. By limiting the acion space, AMC allows us to arrive at the target compression ratio more accurately. The authors empirically observe that Error is inversely proportional to $log(FLOP)$, based on which they frame the reward function for the agent. Thus, this provides a small incentive for reducing FLOPs or the model size, and helping the agent automatically find the limit of compression.

# Experimentation

The authors propose an algorithm to predict the sparsity ratio for all the layers with constrained-model size using fine-grained pruning. The maximum sparsity ratio fine-grained pruning, for the convolutional layers used by the authors was 0.8, and those for fully connected layers was 0.98. Furthermore, for channel pruning, the authors use max response selection, with an actor network containing two hidden layers both with 300 hidden nodes. The actor network first explores 100 episodes with a constant noise of 0.5, and then exploits 300 episodes with exponentially decayed noise.

The AMC is extensively experimented with CIFAR-10 dataset containing 50K training, 10K testing 32*32 images belonging to ten classes. The authors conducted FLOPs-constrained experiments with channel pruning on the CIFAR-10 dataset. In Table 2, it can be seen that AMC outperforms all other deep, shallow and uniform handcrafted model-compression techniques. The uniform policy follows by setting the same compression ratio for each layer uniformly, whereas the shallow and eep policies aggressively prune shallow and eep layers respectively. Furthermore, the authors test the accuracy-guaranteed compression by using the $R_{Param}$ reward. The authors compress ResNet-50 eith fine-grained pruning on CIFAR-10 dataset, and achieve up to 60% compression ratio with even a little higher accuracy on both validation and test set.

The authors also test this methodology on the ImageNet dataset, using 3000 images from the training set to evaluate the reward function, with 224 * 224 input throughout the experiments. In order to achieve sparsity in both activations and weights, the authors use a fine-grained pruning method which prunes neural networks based on individual connections. The problem requires iterative prune and fine-tune procedures, for which the authors conduct 4-iteration pruning and fine-tuning experiments with overall density of full model set to 50%, 35%, 25% and 20% respectively. Due to the higher redundancy, the reinforcement learning agent automatically learns to prune 3*3 convolutional layers with larger sparsity. For this particular dataset, the authors compare the performance of the AMC with existing state-of-the-art channel reduction methods such as FP, RNP and SPP, all of which propose a heuristic strategy to design the prune ratio of each layer. SPP applies PCA analysis to each layer and takes the reconstruction error as the sensitivity measure to determine pruning ratios. In Figure 5, the authors compare the accuracy and MAC among AMC methodology, human expert, and the unpruned MobileNet-v1, wherein AMC dominates human expert., significantly improving the pareto curve of MobileNet-V1.

The AMC technique can optimize FLOPs, model size, and can also optimize the inference latency directly benefiting mobile developers. In Table 4, the authors record the performance speed ups of MobileNet-V1 on pruning with AMC. In comparison, there is a performance degradation when hand-crafted policy is used to prune MobileNet-V1. In Table 5, the authors record the mAP on copressing Faster R-CNN with VGG-16. It is observed that AMC also results in better performance under the same compression ratio on the object detection task. Furthermore, the authors evaluate the generalization ability of AMC on the PASCAL VOC object detection task, achieving 0.7% better mAP than the best hand-crafted pruning methods, thus proving to serve as an effective regularization.

## Strengths & Weaknesses

Prior to this work, the state-of-the-art techniques were hand-crafted, which required domain experts to explore the large design space so as to trade off between model size, speed and accuracy. This is one of the reasons why the biggest strengths of AMC is the automatic design space minimization, without affecting accuracy and resulting in a speed-gain. Another advantage of AMC is that it can be used on a wide variety of networks, and the compression for these networks takes much lesser time than hand-crafter compression techniques. One of the weaknesses of AMC is that it focuses on pruning out unimportant filters instead of creating novel representations. [1]

## Further Work

Wang et al [2] propose Hardware-Aware Automated Quantization technique which relies on generating direct feedback signals from the hardware simulator for the reinforcement agent, rather than using proxy signals like FLOPs and model size. This framework explores the automated quantization of network weights and activations by taking into consideration the hardware architectures. Though structural pruning method works well for CNNs, resource-constrained pruning is a difficult problem since deciding which filter to disable is a NP=hard combinatorial problem. In [3], Chin et al propose layer-compensated pruning where meta-learning is involved to determine better solutions.

## References

[1] Singh, Pravendra, et al. "Leveraging filter correlations for deep model compression." arXiv preprint arXiv:1811.10559 (2018).

[2] Wang, Kuan, et al. "HAQ: Hardware-Aware Automated Quantization with Mixed Precision." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019.

[3] Chin, Ting-Wu, Cha Zhang, and Diana Marculescu. "Layer-compensated pruning for resource-constrained convolutional neural networks." arXiv preprint arXiv:1810.00518 (2018).