MobileOne: An Improved One millisecond Mobile Backbone

Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, Anurag Ranjan

Abstract

In the paper, the authors introduce MobileOne, a novel neural network architecture designed for mobile devices that achieves state-of-the-art accuracy on image classification while running within 1 millisecond. When they deployed on iPhone12, variants of it achieved 75.9% top-1 accuracy on ImageNet. The authors analyze performance bottlenecks in activations and branching that incur high latency costs on mobile in recent efficient networks and propose train-time re-parameterizable branches and dynamic relaxation of regularization in training to alleviate optimization bottlenecks encountered when training small models. The authors also show that MobileOne generalizes well to other tasks such as object detection and semantic segmentation while outperforming recent state-of-the-art efficient models.

Introduction

The paper [1] focuses on the increasing demand for efficient neural networks that can run on mobile devices. Deep learning architectures for mobile devices have seen a lot of progress, with FLOPs and parameter count decreasing while improving accuracy. However, efficiency metric like FLOPs and parameter count may not correlate well with latency when deployed on a mobile device. The authors deployed neural networks on an iPhone12 and benchmark their latency costs. The authors also analyze performance bottlenecks in recent efficient networks and provide ways to mitigate these bottlenecks. The authors conclude by stating that their model generalizes well to other tasks such as object detection and semantic segmentation while outperforming recent state-of-the-art efficient models.

MobileOne introduces linear branches during training that are re-parameterized during inference. However, what sets MobileOne apart from prior works on structural re-parameterization is the addition of trivial over-parameterization branches, which further improves the model's performance in low parameter regimes and model scaling strategies. During inference, MobileOne has a simple feed-forward structure without any branches or skip-connections. This structure incurs lower memory access costs, allowing for wider layers to be incorporated into the network, which boosts the model's representation capacity. MobileOne achieves significant improvements in latency compared to efficient models in literature while maintaining the accuracy on several tasks - image classification, object detection, and semantic segmentation. Furthermore, MobileOne models generalize to other tasks like object detection, and outperform models like MobileNetV3-L and MixNet-S. MobileOne is a novel architecture that runs within 1 ms on a mobile device and achieves state-of-the-art accuracy on image classification within efficient model architectures. The performance of their model also generalizes to a desktop CPU and GPU.

Methodology

1. Metric Correlations

The author converted PyTorch implementation of recent neural networks into ONNX format and then converted them to coreml packages using Core ML Tools and developed an iOS application to measure the latency of the models on iPhone12.
They plotted latency vs. FLOPs and latency vs. parameter count and find that models with higher parameter count have lower latency. Furthermore, convolutional models have lower latency for

similar FLOPs and parameter count than transformer counterparts. They also estimated the Spearman rank correlation. They found latency as moderately correlated with FLOPs and weakly correlated with parameter counts for efficient architectures on a mobile device.

2. Key Bottlenecks

They constructed a 30 layer convolutional neural network and benchmark it on iPhone12 using different activation functions. They used 30 layers based on their analysis of the trade-off between accuracy and latency. The latencies are drastically different due to synchronization costs incurred by recently introduced activation functions like SE-ReLU and Dynamic Shift-Max. Although there is significant accuracy improvement in extremely low FLOP models like MicroNet with DynamicReLU and Dynamic Shift-Max, they have significant latency cost. Hence, they used only ReLU activations in MobileOne.

Architectural blocks like skip connections and squeeze-excite blocks affect runtime performance. Using an architecture with no branches at inference results in smaller memory access cost and limiting the use of squeeze-excite blocks to the biggest variant improves accuracy.

3. MobileOne Architecture

The MobileOne model architecture is based on the evaluation of different design choices where the train-time and inference time architecture are different. During training, they incorporated linear branches that are re-parameterized during inference. This is a key feature that sets apart from prior works on structural re-parameterization. They also included trivial over-parameterization branches, which further improves the performance in low parameter regimes and model scaling strategies.

The basic block builds on the MobileNet-V1 block of 3x3 depth wise convolution followed by 1x1 pointwise convolutions. They then introduced re-parameterizable skip connections with batch normalization along with branches that replicate the structure. These branches are designed to provide further accuracy gains. The trivial over-parameterization factor k is a hyper parameter which is varied from 1 to 5.

During inference, they used a simple feed-forward structure without any branches or skip-connections. This structure incurs lower memory access costs, allowing for wider layers to be incorporated into the network, which boosts my representation capacity.

To better understand the improvements from using train time re-parameterizable branches, they ablated over the versions of MobileOne models by removing train-time re-parameterizable branches while keeping all other training parameters same. Using re-parameterizable branches significantly improves performance. To understand the importance of trivial over-parameterization branches, they ablated over the choice of over-parameterization factor k. For larger variants of MobileOne, the improvements from trivial over-parameterization starts diminishing. For smaller variant like MobileOne-S0, they observed improvements of 0.5% by using trivial over-parameterization branches. They have found that adding re-parameterizable branches improves optimizations as both train and validation losses are further lowered.

Recent works scale model dimensions like width, depth, and resolution to improve performance. MobileOne has similar depth scaling as MobileNet-V2, uses shallower early stages, introduces 5 different width scales, and does not explore scaling up of input resolution as both FLOPs and memory consumption increase. In all experiments, they used cosine schedule for learning rate and progressive learning curriculum for annealing weight decay coefficient and found that annealing the weight decay coefficient gives a 0.5% improvement.

As there is no command line access or functionality on the iPhone 12 to reserve all of a compute fabric for just the model execution. To measure latency on the iPhone 12, they developed an iOS application using swift that runs the models using Core ML. The app runs the models many times and statistic are accumulated to achieve lowest latency and highest consistency. They reported the full round-trip latency for all models, including platform processes that are not model execution. They filtered out interrupts from other processes, and reported the median latency value out of 100 runs.

Experiments

In this section, the authors discussed the experimental evaluation of their performance on various tasks, including image classification, object detection, and semantic segmentation.

In the image classification task, the authors evaluated performance on the ImageNet-1k dataset which consists of 1.28 million training images and a validation set with 50,000 images from 1,000 classes and compared it with other state-of-the-art models. All models are trained for 300 epochs with an effective batch size of 256 using SGD with momentum optimizer. For smaller variants of MobileOne, i.e. S0 and S1, they used standard augmentation – random resized cropping and horizontal flipping. They also used EMA (Exponential Moving Average) weight averaging with decay constant of 0.9995 for training all versions of MobileOne. Current state-of-the-art MobileFormer attains top-1 accuracy of 79.3% with a latency of 70.76ms, while MobileOne-S4 attains 79.4% with a latency of only 1.86ms which is ∼38× faster on mobile. MobileOne-S3 has 1% better top-1 accuracy than EfficientNet-B0 and is faster by 11% on mobile. Their models have a lower latency even on CPU and GPU compared to competing methods. Efficient models are often distilled from a bigger teacher model to further boost the performance. In fact, MobileOne-S4 outperformed even ResNet-50 model which has 72.9% more parameters. MobileOne-S0 has 0.4M less parameters at inference than MobileNetV3-Small and obtains 2.8 percent better top-1 accuracy on ImageNet-1k dataset.

During the Object detection on MS-COCO, they used it as the backbone feature extractor for a single shot object detector SSD to demonstrate the versatility of MobileOne. The model is trained using the mmdetection library on the MS COCO dataset. Their best model outperformed MNASNet by 27.8% and best version of MobileViT by 6.1%.

In the semantic segmentation task, they used MobileOne as the backbone for a Deeplab V3 segmentation network using the cvnets library. Their models outperform Mobile ViT by 1.3% and MobileNetV2 by 5.8% for VOC and ADE 20k, respectively. MobileOne outperforms other efficient architectures significantly on out-of-distribution benchmarks like ImageNet-R and ImageNet-Sketch, and is less robust to corruption when compared to MobileNetV3-L, but outperforms MobileNetV3-L on out-of-distribution benchmarks like ImageNet-C. MobileOne micro architectures are extremely efficient in terms of FLOPS and parameter count, but have similar latency as previous state-of-the-art micro architectures. They also have significantly lower parameter count and better top-1 accuracy.

Conclusion

Overall, they have proposed an efficient, general-purpose backbone for mobile devices that is suitable for image classification, object detection and semantic segmentation. The backbone achieves state-of-the-art performance while being efficient both on a mobile device and a desktop CPU. The architecture and method are designed to provide efficient neural network backbones for mobile devices, with a focus on latency and performance. The authors have extensively analysed different metrics and identified bottlenecks in recent efficient neural networks, resulting in a novel architecture that achieves state-of-the-art performance while being many times faster on mobile.

Although their models were state-of-the-art within the regime of efficient architectures, their accuracy lags large models. They mentioned about aiming at improving the accuracy of these lightweight models on future.

Key Highlights

- The paper addresses the need for neural network architectures that are optimized for deployment on mobile devices, which often have limited computational resources compared to desktop or server machines.
- It highlights the challenge of using traditional metrics like FLOPs (Floating-Point Operations) or parameter count to measure the efficiency of neural networks on mobile devices. These metrics may not accurately reflect the real-world latency and performance of networks on mobile hardware.
- The paper identifies architectural and optimization bottlenecks in existing efficient neural network architectures. These bottlenecks are aspects of the models that limit their performance when deployed on mobile devices.
- The authors introduce MobileOne as a novel neural network backbone designed to address these bottlenecks. They offer different variants of MobileOne, with the fastest variant achieving an impressive inference time of under 1 millisecond on an iPhone 12 while maintaining a high top-1 accuracy rate on the ImageNet dataset.
- MobileOne is claimed to achieve state-of-the-art performance within the category of efficient neural network architectures while being significantly faster when deployed on mobile devices compared to other models.
- The paper provides performance comparisons with other well-known architectures like MobileFormer and EfficientNet, showcasing MobileOne's superior speed and similar or better accuracy.
- MobileOne's advantages are shown to extend to various computer vision tasks, including image classification, object detection, and semantic segmentation. It offers improved latency and accuracy compared to existing efficient architectures when deployed on mobile devices.

Reference Paper

[1] Vasu, P. K. A., Gabriel, J., Zhu, J., Tuzel, O., & Ranjan, A. (2023). MobileOne: An Improved One millisecond Mobile Backbone. arXiv [Cs.CV]. Retrieved from http://arxiv.org/abs/2206.04040