TinyML and Efficient Deep Learning

Song Han

Massachusetts Institute of Technology
Cloud AI → Mobile AI

What’s the next?

Cloud AI

- Memory: 16GB / 32GB
- Computation: 10TFLOPs
- Abundant resources
- Highly accurate

Mobile AI

- Memory: 8GB RAM
- Computation: 100GFLOPs
- Limited resources
- Accuracy-efficiency Tradeoff

Mobile Networks:
- v1, v2, v3, ...
- ShuffleNets: v1, v2, ...
- MnasNet, ProxylessNAS, FBNet, ...

Abundant resources:
- AlexNet, VGGNet,
- Inception, ResNet,
- DenseNet, NASNet,
- AmoebaNet,
- EfficientNet ...

Limited resources:
- MobileNets: v1, v2, v3, ...
- Cloud AI

- Memory: 16GB / 32GB
- Computation: 10TFLOPs
- Abundant resources
- Highly accurate

Cloud AI

- Memory: 8GB RAM
- Computation: 100GFLOPs
- Limited resources
- Accuracy-efficiency Tradeoff

Mobile AI

- Memory: 16GB / 32GB
- Computation: 10TFLOPs
- Abundant resources
- Highly accurate

Cloud AI

- Memory: 8GB RAM
- Computation: 100GFLOPs
- Limited resources
- Accuracy-efficiency Tradeoff

Mobile AI
TinyML: Billions of Connected Intelligence Devices

- Always on
- Privacy preserving
- Extremely limited resources
- Specialized hardware

The number of connected intelligence devices grows rapidly
23.14 billions till 2018
Cloud AI → Mobile AI → Tiny AI

Cloud AI
- Memory: 16GB / 32GB
- Computation: 10TFLOPs
- Abundant resources
- Highly accurate

Mobile AI
- Memory: 8GB RAM
- Computation: 100GFLOPs
- Limited resources
- Accuracy-efficiency Tradeoff

Tiny AI (AIoT)
- Memory: 100 KB-1MB
- Computation: MFLOPs/s
- Always on
- Privacy preserving
- Extremely limited resources
- Highly efficient

Net, how to design it?

- AlexNet, VGGNet,
- Inception, ResNet,
- DenseNet, NASNet,
- AmoebaNet,
- EfficientNet ...
Mobile AI v.s. Tiny AI

Memory: ~GB

→ Soft constraints on 
#weights and #activations

→ vitamins

Small model is good to have :)

Memory: ~KB

→ Hard constraints on 
#weights and #activations

→ pain killer

Small model is a must! (pain killer)
TinyML Techniques

**Model Compression**

- Pruning, NIPS’15
- Deep Compression, ICLR’16, BP
- EIE, ISCA’16
- ESE, FPGA’17, BP
- GAN Compression, CVPR’20

**AutoML & NAS**

- ProxylessNAS, ICLR’19
- HAQ, CVPR’19, oral
- AMC, ECCV’18
- Once-for-All, ICLR’20
- APQ, CVPR’20

1st place, Low Power Computer Vision Challenge’19
1st place, Low Power Computer Vision Challenge’20
1st place, MicroNet Challenge@NeurIPS’19
1st place, Visual Wake Words Challenge@CVPR’19
Model Compression

Large Models → Model → Hardware

Small Models

Han et al, NIPS’15, ICLR’16
GAN requires 10x-500x more computation than image classification!

Conditional GANs are Even More Computationally-Intensive.

<table>
<thead>
<tr>
<th>Model</th>
<th>MACs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MobileNet</td>
<td>0.5G</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>4.0G</td>
</tr>
<tr>
<td>CycleGAN</td>
<td>57G</td>
</tr>
<tr>
<td>GauGAN</td>
<td>281G</td>
</tr>
</tbody>
</table>

Pix2pix, CVPR’17

GAN Compression, CVPR’20
GAN Compression Reduces the Computation by 9-21x

- **Original Model**
- **GAN Compression**

### Horse → zebra
CycleGAN, Zhu et al.
- **Original Model**: 57G
- **GAN Compression**: 2.7G
  - **Reduction**: 21x

### Edge → shoes
Pix2pix, Isola et al.
- **Original Model**: 57G
- **GAN Compression**: 4.8G
  - **Reduction**: 12x

### Cityscapes
GauGAN, Park et al.
- **Original Model**: 281G
- **GAN Compression**: 31.7G
  - **Reduction**: 9x

GAN Compression, CVPR'20
Accelerating Horse2zebra by GAN Compression

Original CycleGAN; MACs: 56.8G; FPS: 12.1; FID: 61.5

GAN Compression; MACs: 3.50G (16.2x); FPS: 40.0 (3.3x); FID: 53.6

Measured on NVIDIA Jetson Xavier GPU
Lower FID indicates better Performance.
Pruning / Deep Compression Adopted in Industry

- **DeePhi Tech / Xilinx** (FPGA and Vitis AI SDK)
- **Samsung** NPU (sparsity-aware, save both energy and cycles)
- **NVIDIA** DLA (sparsity-aware, save energy)
- **Intel** NNP-I (sparsity-aware, save energy)
- **Qualcomm** AIMET (a model efficiency tool) is OS soon.
- **Tensorflow / Keras**
TinyML Techniques

Model Compression

- **Pruning**, NIPS’15
- **Deep Compression**, ICLR’16, BP
- **EIE**, ISCA’16
- **ESE**, FPGA’17, BP
- **GAN Compression**, CVPR’20

AutoML & NAS

- **ProxylessNAS**, ICLR’19
- **HAQ**, CVPR’19, oral
- **AMC**, ECCV’18
- **Once-for-All**, ICLR’20
- **APQ**, CVPR’20
From Manual Design to Automatic Design

Use Human Expertise

Manual Architecture Search

AlexNet, VGGNet, ResNet, MobileNet...


Use Machine Learning

Automatic Architecture Search

# layers
# channels
# kernel
resolution
connectivity
AutoML: Design Automation for Neural Networks

AMC: AutoML for Model Compression

[ICLR 2019]

1st place, Low Power Computer Vision Challenge’19
1st place, Low Power Computer Vision Challenge’20
1st place, MicroNet Challenge@NeurIPS’19
1st place, Visual Wake Words Challenge@CVPR’19

Proxyless Neural Architecture Search

[ICLR 2019]

HAQ: Hardware-aware Automated Quantization

[CVPR 2019], oral
AutoML Outperforms Human Designed Models

Under a wide variety of constraints

Model Size Constrained

Latency

Energy

HAQ: Hardware-Aware Automated Quantization with Mixed Precision
We need Green AI:
Solve the Environmental Problem of NAS

TinyML comes at the cost of BigML
(inference) (training/search)

Problem:

Common carbon footprint benchmarks
in lbs of CO2 equivalent

- Roundtrip flight b/w NY and SF (1 passenger): 1,984
- Human life (avg. 1 year): 11,023
- American life (avg. 1 year): 36,156
- US car including fuel (avg. 1 lifetime): 126,000
- Transformer (213M parameters) w/ neural architecture search: 626,155
- Ours: 52

Evolved Transformer
4 orders of magnitude

ICML’19, ACL’19

“Hardware-Aware Transformer”

Chart: MIT Technology Review • Source: Strubell et al. • Created with Datawrapper
Our Solution (GreenAI & TinyML)

**ONCE-FOR-ALL: TRAIN ONE NETWORK AND SPECIALIZE IT FOR EFFICIENT DEPLOYMENT**

Han Cai\(^1\), Chuang Gan\(^2\), Tianzhe Wang\(^1\), Zhekai Zhang\(^1\), Song Han\(^1\)

\(^1\)Massachusetts Institute of Technology, \(^2\)MIT-IBM Watson AI Lab
{hancai, chuangg, songhan}@mit.edu

**ABSTRACT**

We address the challenging problem of efficient inference across many devices and resource constraints, especially on edge devices. Conventional approaches either manually design or use neural architecture search (NAS) to find a specialized neural network and train it from scratch for each case, which is computationally prohibitive (causing CO\(_2\) emission as much as 5 cars’ lifetime Strubell et al. (2019)) thus unscaleable. In this work, we propose to train a once-for-all (OFA) network that supports diverse architectural settings by decoupling training and search, to reduce the cost. We can quickly get a specialized sub-network by selecting from the OFA network without additional training. To efficiently train OFA networks, we also propose a novel progressive shrinking algorithm, a generalized pruning method that reduces the model size across many more dimensions than pruning (depth, width, kernel size, and resolution). It can obtain a surprisingly large number of subnetworks (> 10\(^10\)) that can fit different hardware platforms and latency constraints while maintaining the same level of accuracy as training independently. On diverse edge devices, OFA consistently outperforms state-of-the-art (SOTA) NAS methods (up to 4.0\% ImageNet top1 accuracy improvement over MobileNetV3, or same accuracy but 1.5× faster than MobileNetV3, 2.6× faster than EfficientNet w.r.t measured latency) while reducing many orders of magnitude GPU hours and CO\(_2\) emission. In particular, OFA achieves a new SOTA 80.0\% ImageNet top-1 accuracy under the mobile setting (<600M MACs). OFA is the winning solution for the 3rd Low Power Computer Vision Challenge (LPCVC), DSP classification track and the 4th LPCVC, both classification and detection track. Code and 50 pre-trained models (for many devices & many latency constraints) are released at https://github.com/mit-han-lab/once-for-all.
Once-for-All Network: Decouple Model Training and Architecture Search

once-for-all network
Once-for-All Network: Decouple Model Training and Architecture Search
Once-for-All Network: Decouple Model Training and Architecture Search
Once-for-All Network: Decouple Model Training and Architecture Search

once-for-all network

Once-for-All, ICLR’20
OFA: 80% Top-1 Accuracy on ImageNet

- Once-for-all sets a new state-of-the-art **80% ImageNet top-1 accuracy** under the mobile vision setting (< 600M MACs).

*Once-for-All*, ICLR'20
OFA Enables Fast Specialization on Diverse Hardware Platforms

**OFA vs. MobileNetV3 vs. MobileNetV2**

- **Samsung S7 Edge Latency (ms)**
- **Google Pixel2 Latency (ms)**
- **LG G8 Latency (ms)**
- **NVIDIA 1080Ti Latency (ms)**
- **Intel Xeon CPU Latency (ms)**
- **Xilinx ZU3EG FPGA Latency (ms)**

**Top-1 ImageNet Acc (%)**

- **SAMSUNG**
- **Google**
- **LG**
- **NVIDIA**
- **Intel**
- **XILINX**

**Batch Size**

- Samsung S7 Edge: 64
- Google Pixel2: 32
- LG G8: 32
- NVIDIA 1080Ti: 64
- Intel Xeon CPU: 1
- Xilinx ZU3EG FPGA: 1 (Quantized)

---

**Once-for-All**, ICLR’20
Our NAS Research Adopted by Industry

- Available at Amazon AutoGluon [1]
- Available at Facebook Pytorch [2]

How to Use ENAS/ProxylessNAS in Ten Minutes

What is the Key Idea of ENAS and ProxylessNAS?

Traditional reinforcement learning-based neural architecture search learns an architecture controller by iteratively sampling the architecture and training the model to get final reward to update the controller. It is extremely expensive process due to training CNN.

Recent work of ENAS and ProxylessNAS construct an over-parameterized network (supernet) and share the weights across different architecture to speed up the search speed. The reward is calculated every few iterations instead of every training period.
TinyML Techniques

Model Compression

AutoML & NAS

1st place, Low Power Computer Vision Challenge’19
1st place, Low Power Computer Vision Challenge’20
1st place, MicroNet Challenge@NeurIPS’19
1st place, Visual Wake Words Challenge@CVPR’19
References

Model Compression & NAS
- **Once-For-All**: Train One Network and Specialize It for Efficient Deployment, ICLR’20
- **ProxylessNAS**: Direct Neural Architecture Search on Target Task and Hardware, ICLR’19
- **APQ**: Joint Search for Network Architecture, Pruning and Quantization Policy, CVPR’20
- **HAQ**: Hardware-Aware Automated Quantization with Mixed Precision, CVPR’19, oral
- **Defensive Quantization**: When Efficiency Meets Robustness, ICLR’19
- **AMC**: AutoML for Model Compression and Acceleration on Mobile Devices, ECCV’18

Efficient Vision:
- **GAN Compression**: Learning Efficient Architectures for Conditional GANs, CVPR’20
- **TSM**: Temporal Shift Module for Efficient Video Understanding, ICCV’19
- **PVCNN**: Point Voxel CNN for Efficient 3D Deep Learning, NeurIPS’19, spotlight

Efficient NLP:
- **Lite Transformer** with Long Short Term Attention, ICLR’20
- **HAT**: Hardware-aware Transformer, ACL’20

Hardware & EDA:
- **SpArch**: Efficient Architecture for Sparse Matrix Multiplication, HPCA’20
- **Transferable Transistor Sizing** with Graph Neural Networks and Reinforcement Learning, DAC’20
HAN Lab Students: Yujun Lin (Arch PhD), Hanrui Wang (Arch PhD), Zhijian Liu (ML PhD)

The pareto frontier of such trade-offs is a co-design of number representation together with model complexity arrays. I plan to use machine learning techniques to find the best number representation for deep learning. The design space includes:

- [training, inference] x [channel number] x [layer number] x [bit width] x [decimal point]
- [linear quantization, log quantization, kmeans quantization] x

For deep learning, the design space is large and hard to be explored by human. It should be explored carefully to avoid underflow or overflow [NVIDIA'17]. Given the large extremes of quantization, the latter has easy hw implementation but poor expressiveness. The former has hard hw implementation (need register lookup) but it lacks flexibility. TACO [Kjolstad'17] is a flexible compiler for sparse linear algebra on CPU, but it lacks accelerator support. Therefore, I plan to work on an specialized accelerator for sparse machine learning.

Pruning techniques [Han'15] show that DNN models can be pruned to very sparse, saving the FLOPs by 10x and model size by 8x (FC layer, index included). However, it's challenging for general purpose hardware to take advantage of sparsity. EIE [Han'16] is the first hardware accelerator for sparse DNN, it's efficient but it lacks flexibility. TACO is generously sponsored by:

- IBM
- Texas Instruments
- Maxim Integrated
- Intel
- Qualcomm
- DARPA
- NSF
- Facebook
- Google
- AWS
- Sony
- Xilinx
- Samsung
- Hyundai