tinyML Summit

Miniature dreams can come true...

March 28-30, 2022 | San Francisco Bay Area

www.tinyML.org
TinyML for All: Full-stack Optimization for Diverse Edge AI Platforms

Di Wu
Co-founder and CEO, OmniML

Song Han
Assistant Professor, MIT EECS
Co-founder and Chief Scientist, OmniML
TinyML is about Constraints

Mismatch: AI has been evolving unconstrained for many years

<table>
<thead>
<tr>
<th></th>
<th>Cloud AI</th>
<th>Mobile AI</th>
<th>Tiny AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation</td>
<td>10 TFLOPS</td>
<td>GFLOPS</td>
<td>MFLOPS</td>
</tr>
<tr>
<td>Memory</td>
<td>32GB</td>
<td>4GB</td>
<td>256KB</td>
</tr>
</tbody>
</table>

100,000x smaller
Everything Together: Real-world AI on Tiny MCUs


Facemask Detection

Person Detection

Works on Cortex M7 MCU
Brief History of MCUNets

Reducing the model sizes with increasing accuracy

[MCUNet, NeurIPS'20]
[MCUNet-v2, NeurIPS'21]
Opportunity in Fundamental ML Algorithms
Making algorithm more efficient under existing constraints

Faster than Moore’s Law:
3.5x model size reduction every 12 months

Improving efficiency means *more accurate* models, too

TinyML is about improving the entire stack: from design to deployment, from computation to data

[MCUNet, NeurIPS'20]
[MCUNet-v2, NeurIPS'21]
Agenda

Focus on Constraints on the Entire Stack

Data Collection → Model Design → Training → Deployment → Maintain

MCUNet-v1, MCUNet-v2

Once-for-All

TinyGAN  PVCNN  NetAug  TorchSparse  TinyTL
MCUNet-v1: TinyNAS+TinyEngine Co-design

TinyNAS:
• Re-design the design space
• Latency-aware
• Energy-aware
• Once-for-all Network

TinyEngine:
• Co-design, specialization
• Offload run-time to compile-time
• Graph optimizations
• Memory-aware scheduling
• Low-precision
• Assembly-level optimizations

AutoML, Efficient Neural Architecture

Efficient Compiler / Runtime

[MCUNet, NeurIPS’20]
New Problem: Imbalanced Memory Distribution of CNNs [MCUNet-v2, NeurIPS'21]

Per-block memory usage of MobileNetV2
Solving the Imbalance with Patch-based Inference

After applying Patch-based Inference

[MCUNet-v2, NeurIPS'21]
Solving inference bottleneck (peak memory) results in smaller and better models
Agenda

Focus on Constraints on the Entire Stack

- Data Collection
- Model Design
- Training
- Deployment
- Maintain

- MCUNet-v1, MCUNet-v2
- TinyGAN
- PVCNN
- NetAug
- TorchSparse
- TinyTL

Once-for-All
Once-for-All Network
Train once, get many; Fit diverse hardware constraints
Better Results with Much Smaller Training Cost
Reduce the search cost from 42,000 GPU hours (Google) to 200 GPU hours

Existing: Lots of hand tuning for different devices and latency.
OFA: Auto design the NN architecture at low cost
Agenda

Focus on Constraints on the Entire Stack

Data Collection → Model Design → Training → Deployment → Maintain

- MCUNet-v1, MCUNet-v2
- Once-for-All
- DiffAugment
- PVCNN
- NetAug
- TorchSparse
- TinyTL
Problem in Training for Tiny Models

Existing Training Techniques don't Apply to TinyML

[NetAug, ICLR'22]
NetAug for TinyML

Augment Model Rather than Data

1. Input
2. Build an Augmented Model

Zero Inference Overhead

[NetAug, ICLR'22]
Agenda
Focus on Constraints on the Entire Stack

[TinyTL, NeurIPS'20]
Problem: Training Memory is much Larger
Bottleneck is Activation rather than Parameters

[TinyTL, NeurIPS'20]
TinyTL: Up to 6.5x Memory Saving without Accuracy Loss

Use Fine-Tune Bias Only and Lite Residual Learning

[TinyTL, NeurIPS'20]
Agenda

Focus on Constraints on the Entire Stack

- Data Collection
- Model Design
- Training
- Deployment
- Maintain

MCUNet-v1, MCUNet-v2
Once-for-All
TinyGAN
PVCNN
NetAug
TorchSparse
TinyTL

[DiffAugment, NeurIPS'20]
Data is Also Constrained

Many TinyML Applications Have Limited Access to Data

Rare Defects

Specific Tasks

Privacy Concerns

[DiffAugment, NeurIPS'20]
Differentiable Augmentation

Photo-realistic and Smooth Generation with 100 Training Images

[DiffAugment, NeurIPS'20]
Agenda

Focus on Constraints on the Entire Stack

- Data Collection
- Model Design
- Training
- Deployment
- Maintain

MCUNet-v1, MCUNet-v2

Once-for-All

DiffAugment
PVCNN
NetAug
TorchSparse
TinyTL
TinyML for LIDAR & Point Cloud

Challenge: High Algorithm Complexity vs. Limited Computational Resource

[PCVNN, NeurIPS'19]
[SPVNAS, ECCV'20]
[PointAcc, Micro’21]
[TorchSparse, MLSys’22]
while (c) and (d) are emerging e
Takeaways: Coming Back to MCUNets

Co-optimization on the entire stack is the key to unlock the most potential for TinyML
Effortlessly Empower Edge AI Everywhere
Fundamental Problems in TinyML

ML under new HW constraints is very hard

Typical AI/ML Development

Data Collection → Model Design → Training → Deployment → Maintain

Designing new models that works on different HW is still a manual and iterative approach

Mismatch

Slow Adoption

Less Revenue/Volume
OmniML “Compress” the Model Before Training

Bring HW deployment constraints into model design and training
OmniML: Enable TinyML for All Vision Tasks
Create the Best Models on Different Platforms Effortlessly

CV on Mobile Devices
- Pose estimation
- Scene Segmentation
- Image denoise, super resolution
- AR/VR

Smarter Cameras
Turn “dumb” cameras into AI-powered cameras with advanced CV features on low-power, low-cost chip.

Sensor Fusion
3D Detection
Multi-sensor 3D object detection for automotive applications.

Computer Vision on MCUs
Not only classification but also object detection on microcontrollers with only 256~512KB of memory.

40+ Customers Conversations
10+ POCs
100K Installed devices
Founding Team

Leading Experts in Efficient Deep Learning

**Song Han**
- Assistant professor at MIT, PhD from Stanford
- Co-founder of DeePhi Tech (acquired by Xilinx)
- “35 Innovators Under 35” by MIT Technology Review
- NSF CAREER Award, IEEE “AI 10 to Watch”
- Inventor of “Deep Compression”
- 29K Google Scholar citations

**Di Wu**
- Previous tech lead at Facebook AI, PyTorch accelerator enablement
- Product and engineering leader at Falcon Computing Solutions (acquired by Xilinx)
- PhD from UCLA, years of experience in customized hardware systems at Intel Lab, MSRA.

**Huizi Mao**
- PhD from Stanford. Co-Inventor of “Deep Compression”
- Early member of DeePhi and Megvii.
- Worked at Google Research, Facebook AML and NVIDIA.
- NVIDIA Fellowship Recipient.
Come talk to us to learn more

Follow us:
https://www.linkedin.com/company/omniml
https://twitter.com/OmniML_AI

We are hiring:
https://omniml.ai/career/
contact@omniml.ai

Thank you
Copyright Notice

This presentation in this publication was presented as a tinyML® Summit 2022. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org