“AutoML for TinyML with Once-for-All Network”

Song Han – MIT
April 28, 2020

www.tinyML.org
tinyML Talks Sponsors

EDGE IMPULSE

Additional Sponsorships available – contact Bette@tinyML.org for info
# Next tinyML Talk

<table>
<thead>
<tr>
<th>Date</th>
<th>Presenter</th>
<th>Topic / Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thursday</td>
<td>Hans Reyserhove</td>
<td>Embedded Computer Vision Hardware through the Eyes of an AR Glass</td>
</tr>
<tr>
<td>May 14</td>
<td>Postdoctoral Research Scientist, Facebook Reality Labs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jamie Campbell</td>
<td>Using TensorFlow Lite for Microcontrollers for High-Efficiency NN Inference on Ultra-Low Power Processors</td>
</tr>
<tr>
<td></td>
<td>Software Engineering Manager, Synopsys</td>
<td></td>
</tr>
</tbody>
</table>

Webcast start time is 8 am Pacific time
Each presentation is approximately 30 minutes in length

Please contact talks@tinyml.org if you are interested in presenting
Song Han is an assistant professor in MIT’s Department of Electrical Engineering and Computer Science. His research focuses on efficient deep learning computing. He has proposed “deep compression” that can reduce neural network size by an order of magnitude, and the hardware implementation “efficient inference engine” that first exploited model compression and weight sparsity in deep learning accelerators. He received a best paper award at the ICLR’16 and FPGA’17. He is a recipient of NSF CAREER Award and MIT Technology Review Innovators Under 35. Many of the pruning, compression, and acceleration techniques have been integrated into commercial AI chips. He was the co-founder and chief scientist of DeePhi Tech that was acquired by Xilinx. He earned a PhD in electrical engineering from Stanford University.
AutoML for TinyML with Once-for-All Network

Song Han
Massachusetts Institute of Technology
AutoML for TinyML with Once-for-All Network

Less Engineer Resources: AutoML
Less Computational Resources: TinyML
Our 1st generation solution

Proxyless Neural Architecture Search
[ICLR 2019]

AMC: AutoML for Model Compression
[ICLR 2019]

REFERENCES


CONCLUSION

...
Challenge: Efficient Inference on Diverse Hardware Platforms

Cloud AI
- Memory: 32GB
- Computation: TFLOPS/s

Mobile AI
- Memory: 4GB
- Computation: GFLOPS/s

Tiny AI (AIoT)
- Memory: <100 KB
- Computation: <MFLOPS/s

- Different hardware platforms have different resource constraints. We need to customize our models for each platform to achieve the best accuracy-efficiency trade-off, especially on resource-constrained edge devices.
Challenge: Efficient Inference on Diverse Hardware Platforms

for training iterations: forward-backward();

Design Cost (GPU hours)

200

The design cost is calculated under the assumption of using MobileNet-v2.
Challenge: Efficient Inference on **Diverse** Hardware Platforms

(1) for search episodes:  //meta controller
   for training iterations:  \texttt{forward-backward();}
      if good_model: break;
   for post-search training iterations:  \texttt{forward-backward();}

Design Cost (GPU hours)

- \texttt{40K}

The design cost is calculated under the assumption of using MnasNet.
Challenge: Efficient Inference on **Diverse** Hardware Platforms

Diverse Hardware Platforms

<table>
<thead>
<tr>
<th>Year</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td></td>
</tr>
<tr>
<td>2017</td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td></td>
</tr>
</tbody>
</table>

Design Cost (GPU hours)

- 40K
- 160K

(1) for search episodes: //meta controller
    for training iterations: forward-backward();
    if good_model: break;

(2) for devices:
    for post-search training iterations: forward-backward();

The design cost is calculated under the assumption of using MnasNet.

Once-for-All, ICLR’20

Challenge: Efficient Inference on Diverse Hardware Platforms

Diverse Hardware Platforms

Cloud AI (10^{12} FLOPS)  Mobile AI (10^9 FLOPS)  Tiny AI (10^6 FLOPS)

(1) for search episodes: //meta controller
for training iterations: forward-backward();
if good_model: break;
for post-search training iterations: forward-backward();

Design Cost (GPU hours)

- 40K
- 160K
- 1600K

The design cost is calculated under the assumption of using MnasNet.
Challenge: Efficient Inference on **Diverse** Hardware Platforms

Diverse Hardware Platforms

Cloud AI ($10^{12}$ FLOPS)  Mobile AI ($10^9$ FLOPS)  Tiny AI ($10^6$ FLOPS)

(2) for many devices:

(1) for search episodes: //meta controller

for training iterations: forward-backward();

if good_model: break;

for post-search training iterations: forward-backward();

Design Cost (GPU hours)

<table>
<thead>
<tr>
<th>Size</th>
<th>Design Cost</th>
<th>CO$_2$ Emission</th>
</tr>
</thead>
<tbody>
<tr>
<td>40K</td>
<td>11.4k lbs</td>
<td>45.4k lbs CO$_2$ emission</td>
</tr>
<tr>
<td>160K</td>
<td>45.4k lbs CO$_2$ emission</td>
<td></td>
</tr>
<tr>
<td>1600K</td>
<td>454.4k lbs CO$_2$ emission</td>
<td></td>
</tr>
</tbody>
</table>

1 GPU hour translates to 0.284 lbs CO$_2$ emission according to Strubell, Emma, et al. "Energy and policy considerations for deep learning in NLP." ACL. 2019.
We need Green AI: Solve the Environmental Problem of NAS

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

Common carbon footprint benchmarks
in lbs of CO2 equivalent

<table>
<thead>
<tr>
<th>Activity</th>
<th>Carbon Footprint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roundtrip flight b/w NY and SF (1 passenger)</td>
<td>1,984</td>
</tr>
<tr>
<td>Human life (avg. 1 year)</td>
<td>11,023</td>
</tr>
<tr>
<td>American life (avg. 1 year)</td>
<td>36,156</td>
</tr>
<tr>
<td>US car including fuel (avg. 1 lifetime)</td>
<td>126,000</td>
</tr>
<tr>
<td>Transformer (213M parameters) w/ neural architecture search</td>
<td>626,155</td>
</tr>
</tbody>
</table>

Ours 52  4 orders of magnitude  ACL’20

Evolved Transformer  ICML’19, ACL’19

Hardware-Aware Transformer

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by Karen Hao  June 6, 2019

The artificial-intelligence industry is often compared to the oil industry: once mined and refined, data, like oil, can be a highly lucrative commodity. Now it seems the metaphor may extend even further. Like its fossil-fuel counterpart, the process of deep learning has an outsize environmental impact.
Our new solution, OFA: Decouple Training and Search

Conventional NAS

(2) for devices:
  (1) for search episodes:
    for training iterations:
      forward-backward();
      if good_model: break;
    for post-search training iterations:
      forward-backward();

=>

Once-for-All:

for OFA training iterations:
  forward-backward();

for devices:
  for search episodes:
    sample from OFA;
    if good_model: break;
    direct deploy without training;

For OFA training iterations:
    forward-backward();

=>

For devices:
  for search episodes:
    sample from OFA;
    if good_model: break;
    direct deploy without training;
Challenge: Efficient Inference on Diverse Hardware Platforms

Diverse Hardware Platforms

Cloud AI ($10^{12}$ FLOPS)
Mobile AI ($10^9$ FLOPS)
Tiny AI ($10^6$ FLOPS)

for OFA training iterations:
  forward-backward();

for devices:
  for search episodes:
    sample from OFA;
    if good_model: break;
    direct deploy without training;

Design Cost (GPU hours)

training
  40K → 11.4k lbs CO$_2$ emission
search
  160K → 45.4k lbs CO$_2$ emission

Once-for-All Network

1600K → 454.4k lbs CO$_2$ emission

1 GPU hour translates to 0.284 lbs CO$_2$ emission according to Strubell, Emma, et al. "Energy and policy considerations for deep learning in NLP." ACL. 2019.
Once-for-All Network:
Decouple Model Training and Architecture Design

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

Once-for-All, ICLR’20
Once-for-All Network: Decouple Model Training and Architecture Design

Once-for-All, ICLR’20
Challenge: how to prevent different subnetworks from interfering with each other?
Solution: Progressive Shrinking

- More than $10^{19}$ different sub-networks in a single once-for-all network, covering 4 different dimensions: resolution, kernel size, depth, width.
- Directly optimizing the once-for-all network from scratch is much more challenging than training a normal neural network given so many sub-networks to support.
Solution: Progressive Shrinking

- More than $10^{19}$ different sub-networks in a single once-for-all network, covering 4 different dimensions: resolution, kernel size, depth, width.
- Directly optimizing the once-for-all network from scratch is much more challenging than training a normal neural network given so many sub-networks to support.

**Progressive Shrinking**

- Small sub-networks are nested in large sub-networks.
- Cast the training process of the once-for-all network as a progressive shrinking and joint fine-tuning process.
Connection to Network Pruning

Network Pruning
- Train the full model
- Shrink the model (only width)
- Fine-tune the small net
- Single pruned network

Progressive Shrinking
- Train the full model
- Shrink the model (4 dimensions)
- Fine-tune both large and small sub-nets
- Once-for-all network

- Progressive shrinking can be viewed as a generalized network pruning with much higher flexibility across 4 dimensions.
Progressive Shrinking

Randomly sample input image size for each batch
Progressive Shrinking

Randomly sample input image size for each batch
Randomly sample input image size for each batch
Progressive Shrinking

Randomly sample input image size for each batch
Randomly sample input image size for each batch

Once-for-All, ICLR’20
Progressive Shrinking

Randomly sample input image size for each batch
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix

- Full Elastic Resolution → Full Elastic Kernel Size → Full Elastic Depth → Full Elastic Width

Transition:
- Full 7x7 to Partial 5x5 to Partial 3x3

Transform Matrix:
- 25x25 for 7x7
- 9x9 for 5x5
- 3x3 for 3x3

Once-for-All, ICLR’20
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix

Elastic Resolution
Full
Partial

Elastic Kernel Size
Full
Partial

Elastic Depth
Full

Elastic Width
Full

7x7
Transform Matrix 25x25

5x5
Transform Matrix 9x9

3x3
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix
Progressive Shrinking

Start with full kernel size
Smaller kernel takes centered weights via a transformation matrix
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth

Once-for-All, ICLR’20
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth.
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth

Once-for-All, ICLR’20
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth.
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth
Progressive Shrinking

Gradually allow later layers in each unit to be skipped to reduce the depth
Progressive Shrinking

Gradually shrink the width

Keep the most important channels when shrinking via channel sorting

Once-for-All, ICLR’20
Progressive Shrinking

Gradually shrink the width

Keep the most important channels when shrinking via channel sorting
### Progressive Shrinking

**Once-for-All, ICLR’20**

<table>
<thead>
<tr>
<th>Elastic Resolution</th>
<th>Full</th>
<th>Partial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic Kernel Size</td>
<td>Full</td>
<td>Partial</td>
</tr>
<tr>
<td>Elastic Depth</td>
<td>Full</td>
<td>Partial</td>
</tr>
<tr>
<td>Elastic Width</td>
<td>Full</td>
<td>Partial</td>
</tr>
</tbody>
</table>

- **Train with full width**
- **Channel sorting**
  - Channel importance:
    - O1: 0.02, 0.15, 0.85, 0.63
- **Progressively shrink the width**
  - Channel sorting:
    - O1: 0.82, 0.11, 0.46
- **Keep the most important channels when shrinking via channel sorting**

**Gradually shrink the width**

---

**Elastic Resolution**

**Elastic Kernel Size**

**Elastic Depth**

**Elastic Width**

**Full**

**Partial**

---

**Reorg.**

-O1 → O2 → O3

-O1 → O2 → O1

---

**Keep the most important channels when shrinking via channel sorting**
Progressive Shrinking

Gradually shrink the width
Keep the most important channels when shrinking via channel sorting

Once-for-All, ICLR’20
Progressive Shrinking

Gradually shrink the width

Keep the most important channels when shrinking via channel sorting

Once-for-All, ICLR’20
Progressive Shrinking

Gradually shrink the width
Keep the most important channels when shrinking via channel sorting

Once-for-All, ICLR’20
Progressive Shrinking

put it together:

Elastic Resolution

R ∈ [128, 132, …, 224]

Train full network

K = 7
D = 4
W = 6

Elastic Resolution

Elastic Kernel Size

K ∈ {7, 5, 3}

Sample K at each layer

Generate kernel weights (Fig. 3)

Fine-tune weights & transformation matrix

Elastic Depth

D ∈ {4, 3, 2}

Sample D at each unit; sample K

Keep the first D layers at each unit (Fig. 3)

Fine-tune weights

W ∈ {6, 4, 3}

Sample W at each layer; sample K, D

Fine-tune weights

Once-for-all Network

Elastic Width

D ∈ {4, 3, 2}
K ∈ {7, 5, 3}

Channel sorting (Fig. 4)

Once-for-All, ICLR’20
Performances of Sub-networks on ImageNet

Sub-networks under various architecture configurations
D: depth, W: width, K: kernel size

- Progressive shrinking consistently improves accuracy of sub-networks on ImageNet.

Once-for-All, ICLR’20
How about search?

```plaintext
for OFA training iterations:
  forward-backward();

for devices:

for search episodes:
  sample from OFA; //with evolution
  if good_model: break;
  direct deploy without training;
```

Once-for-All, ICLR’20
How to evaluate if good_model? — by Model Twin

Acc Dataset
[Architecture, Accuracy]

Accuracy Prediction Model

Latency Dataset
[Architecture, Latency]

Latency Prediction Model

Accuracy/Latency predictor
RMSE ~0.2%

OFA Network

Specialized
Sub-Network

Predictor-based
Architecture Search
2.6x faster than EfficientNet, 1.5x faster than MobileNetV3

- Training from scratch cannot achieve the same level of accuracy
More accurate than training from scratch

- Training from scratch cannot achieve the same level of accuracy

**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

Once-for-All, ICLR’20
We need Green AI
Solve the Environmental Problem of NAS

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

Evolved Transformer

Training a single AI model can emit as much carbon as five cars in their lifetimes

Deep learning has a terrible carbon footprint.

by Karen Hao
June 6, 2019

The artificial-intelligence industry is often compared to the oil industry: once mined and refined, data, like oil, can be a highly lucrative commodity. Now it seems the metaphor may extend even further. Like its fossil-fuel counterpart, the process of deep learning has an outsized environmental impact.
How to save CO$_2$ emission

1. Once for all: **Amortize** the search cost across **many** sub-networks and deployment scenarios

2. Lite-transformer: **Human-in-the-loop** design. Apply human insights of HW&ML, rather than “just search it”

---

**Once-for-All**, ICLR’20

**Lite Transformer**, ICLR’20
OFA has broad applications

- Efficient Transformer
- Efficient Video Recognition
- Efficient 3D Vision
- Efficient GAN Compression
OFA’s Application: Hardware-Aware Transformer

Efficient NLP on mobile devices enable real time conversation between speakers using different languages

"Nice to meet you"
Encantada de conocerte"
“만나서 반갑습니다”
“很高兴见到你"
"Freut mich, dich kennenzulernen"

Human Life (Avg. 1 year) 11,023
American Life (Avg. 1 year) 36,156
US Car w/ Fuel (Avg. 1 lifetime) 126,000
Evolved Transformer 626,155
HAT (Ours) 12041x

3.7x smaller model size, same performance on WMT’14 En-De;
3x, 1.6x, 1.5x faster on Raspberry Pi, CPU, GPU than Transformer Baseline
12,000x less CO₂ than evolved transformer

HAT, ACL’20
OFA’s Application: Efficient Video Recognition

- **7x** less computation, same performance as TSM+ResNet50
- same computation, **3%** higher accuracy than TSM+MobileNet-v2
OFA’s Application: Efficient 3D Recognition

AR/VR: a whole backpack of computer

self-driving: a whole trunk of GPU

4x FLOPs reduction and 2x speedup over MinkowskiNet
3.6% better accuracy under the same computation budget.

followup of PVCNN, NeurIPS’19 (spotlight)
OFA’s Application: GAN Compression

Accelerating Horse2zebra by GAN Compression

<table>
<thead>
<tr>
<th>Metric</th>
<th>CycleGAN</th>
<th>Pix2pix</th>
<th>GauGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>FID (↓)</td>
<td>61.5→65.0</td>
<td>24.2→26.6</td>
<td>–</td>
</tr>
<tr>
<td>mAP (↑)</td>
<td>–</td>
<td>–</td>
<td>58.9→58.4</td>
</tr>
<tr>
<td>MAC Reduction</td>
<td>21.2×</td>
<td>11.8×</td>
<td>8.8×</td>
</tr>
<tr>
<td>Memory Reduction</td>
<td>2.0×</td>
<td>1.7×</td>
<td>1.8×</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>CPU</th>
<th>GPU</th>
<th>CPU</th>
<th>GPU</th>
<th>CPU</th>
<th>GPU</th>
<th>CPU</th>
<th>GPU</th>
<th>CPU</th>
<th>GPU</th>
<th>CPU</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xavier</td>
<td>1.65s (18.5×)</td>
<td>0.026s (3.1×)</td>
<td>3.07s (9.9×)</td>
<td>0.035s (2.4×)</td>
<td>0.10s (3.2×)</td>
<td>21.2s (7.9×)</td>
<td>0.026s (3.1×)</td>
<td>0.035s (2.4×)</td>
<td>0.10s (3.2×)</td>
<td>21.2s (7.9×)</td>
<td>0.026s (3.1×)</td>
<td>0.035s (2.4×)</td>
</tr>
<tr>
<td>Nano</td>
<td>6.30s (14.0×)</td>
<td>0.16s (4.0×)</td>
<td>8.57s (10.3×)</td>
<td>0.26s (2.5×)</td>
<td>0.81s (3.3×)</td>
<td>65.3s (8.6×)</td>
<td>0.16s (4.0×)</td>
<td>0.26s (2.5×)</td>
<td>0.81s (3.3×)</td>
<td>65.3s (8.6×)</td>
<td>0.16s (4.0×)</td>
<td>0.26s (2.5×)</td>
</tr>
<tr>
<td>Speedup</td>
<td>0.005s (2.5×)</td>
<td>0.007s (1.8×)</td>
<td>0.034s (1.7×)</td>
<td>0.005s (2.5×)</td>
<td>0.007s (1.8×)</td>
<td>0.034s (1.7×)</td>
<td>0.005s (2.5×)</td>
<td>0.007s (1.8×)</td>
<td>0.034s (1.7×)</td>
<td>0.005s (2.5×)</td>
<td>0.007s (1.8×)</td>
<td>0.034s (1.7×)</td>
</tr>
<tr>
<td>Speedup</td>
<td>0.11s (3.4×)</td>
<td>0.15s (2.6×)</td>
<td>0.74s (2.8×)</td>
<td>0.11s (3.4×)</td>
<td>0.15s (2.6×)</td>
<td>0.74s (2.8×)</td>
<td>0.11s (3.4×)</td>
<td>0.15s (2.6×)</td>
<td>0.74s (2.8×)</td>
<td>0.11s (3.4×)</td>
<td>0.15s (2.6×)</td>
<td>0.74s (2.8×)</td>
</tr>
</tbody>
</table>

8-21x FLOPs reduction on CycleGAN, Pix2pix, GauGAN

1.7x-18.5x speedup on CPU/GPU & Mobile CPU/GPU

GAN Compression, CVPR’20
Summary: Once-for-All Network

- We introduce once-for-all network for **efficient inference on diverse hardware platforms**.
- We present an effective **progressive shrinking** approach for training once-for-all networks.

**Progressive Shrinking**

- Once-for-all network **surpasses MobileNetV3 and EfficientNet** by a large margin under all scenarios, setting a new state-of-the-art **80% ImageNet Top1-accuracy** under the mobile setting (< 600M MACs).
  - **First place** in the 3rd Low-Power Computer Vision Challenge, DSP track @ICCV’19
  - **First place** in the 4th Low-Power Computer Vision Challenge @NeurIPS’19, both classification & detection.

- Released **50+ different pre-trained OFA models** on diverse hardware platforms (CPU/GPU/FPGA/DSP).
  
  ```python
  net, image_size = ofa_specialized(net_id, pretrained=True)
  ```

- Released the **training code & pre-trained OFA network** that provides diverse sub-networks without training.
  
  ```python
  ofa_network = ofa_net(net_id, pretrained=True)
  ```

Project Page: [https://ofa.mit.edu](https://ofa.mit.edu)
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
- 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

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
Make AI Efficient:
Tiny Computational Resources
Tiny Human Resources

Media Coverage:
Website: songhan.mit.edu

youtube.com/c/MITHANLab

github.com/mit-han-lab

MIT News
MIT Technology Review
Wired
Engadget
IEEE Spectrum
Diverse Hardware Platforms, 50+ Pretrained Models are Released

OFA based on FLOPs
- \text{tops}@656\text{M}\text{top}@80.1\text{finetune}@75
- \text{tops}@482\text{D}\text{top}@79.6\text{finetune}@75
- \text{tops}@389\text{M}\text{top}@79.1\text{finetune}@75

OFA for Mobile Phones

<table>
<thead>
<tr>
<th>Platform</th>
<th>Note8</th>
<th>Note10</th>
<th>Pixel1</th>
<th>Pixel2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LG G8</td>
<td>\text{note8}\text{lat}@843\text{ms}\text{top}@80.2\text{finetune}@75</td>
<td>\text{note10}\text{lat}@856\text{ms}\text{top}@79.3\text{finetune}@75</td>
<td>\text{pixel1}\text{lat}@813\text{ms}\text{top}@79.8\text{finetune}@75</td>
<td>\text{pixel2}\text{lat}@795\text{ms}\text{top}@79.5\text{finetune}@75</td>
</tr>
<tr>
<td>Google Pixel1</td>
<td>\text{pixel1}\text{lat}@832\text{ms}\text{top}@79.3\text{finetune}@75</td>
<td>\text{pixel2}\text{lat}@795\text{ms}\text{top}@79.5\text{finetune}@75</td>
<td>\text{pixel1}\text{lat}@832\text{ms}\text{top}@79.3\text{finetune}@75</td>
<td>\text{pixel2}\text{lat}@795\text{ms}\text{top}@79.5\text{finetune}@75</td>
</tr>
<tr>
<td>Google Pixel2</td>
<td>\text{pixel1}\text{lat}@805\text{ms}\text{top}@79.3\text{finetune}@75</td>
<td>\text{pixel2}\text{lat}@795\text{ms}\text{top}@79.5\text{finetune}@75</td>
<td>\text{pixel1}\text{lat}@805\text{ms}\text{top}@79.3\text{finetune}@75</td>
<td>\text{pixel2}\text{lat}@795\text{ms}\text{top}@79.5\text{finetune}@75</td>
</tr>
</tbody>
</table>

OFA for Desktop (CPUs and GPUs)

<table>
<thead>
<tr>
<th>Platform</th>
<th>V100 GPU</th>
<th>Intel Xeon CPU (Batch Size 1)</th>
<th>NVIDIA Tesla V100 (Batch Size 64)</th>
<th>NVIDIA 1080Ti (Batch Size 64)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jetson TX2 GPU</td>
<td>\text{tx2}\text{gpu}@668\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{tx2}\text{gpu}@668\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{tx2}\text{gpu}@668\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{tx2}\text{gpu}@668\text{ms}\text{top}@78.7\text{finetune}@25</td>
</tr>
<tr>
<td>Xilinx ZU9EG FPGA (Batch Size 1)</td>
<td>\text{z9e}\text{fpga}@631\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{z9e}\text{fpga}@631\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{z9e}\text{fpga}@631\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{z9e}\text{fpga}@631\text{ms}\text{top}@78.7\text{finetune}@25</td>
</tr>
<tr>
<td>Intel Xeon CPU with MLK-DNN (Batch Size 1)</td>
<td>\text{cpu}\text{lat}@797\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{cpu}\text{lat}@797\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{cpu}\text{lat}@797\text{ms}\text{top}@78.7\text{finetune}@25</td>
<td>\text{cpu}\text{lat}@797\text{ms}\text{top}@78.7\text{finetune}@25</td>
</tr>
</tbody>
</table>

Once-for-All, ICLR'20
OFA for FPGA Accelerators

Measured results on XILINX FPGA

- Non-specialized neural networks do not fully utilize the hardware resource. There is a large room for improvement via neural network specialization.

Once-for-All, ICLR’20
Copyright Notice

This presentation in this publication was presented as a tinyML® Talks webcast. 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
Copyright Notice

This presentation in this publication was presented as a tinyML® Talks webcast. 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