Projects: Model Acceleration, Compression, and Optimization

Score: 20% (+6% extra at maximum) [scoring criteria refer to Table-1]

Group members: 2 or 3 persons

Presentation day: Oct 7th, 2024 [W10]

Description: Compete to improve the model with better efficiency, effectiveness, robust, lightweight, and so on, according to the measurement criteria in Table I. This model is implemented with **non-labeled target data in learning scenarios.**

Model reference: *MobileNetV3-large/small (**Labeled** source data) [Optional: Multiple or Single MBV3-l] → MobileNetV3-small (**Unlabeled** target data) (**MBV3-l → MBV3-s**)* [W4-2_Use SOTA model.ipynb]

Baseline Scenario in comparison with: *MobileNetV3-small (**Labeled** target data)*

Weight initialization: Pretrained weight from Torchvision (For both MBV3 large and small models)

Techniques (Guidelines):

- Modifying block layers or new creating sub-modules [W3-W4]
  Ref1: https://github.com/VSainteuf/lightweight-temporal-attention-pytorch
  Ref2: https://github.com/Nandan91/ULSAM
  Ref3: https://github.com/Qrange-group/LSAS
  Ref4: https://github.com/moskomule/senet.pytorch
- Inductive transfer learning [W5], such as
  - Fine-tuning: MBV3-small (labeled source) → MBV3-small (non-labeled target)
- Transductive transfer learning [W5], such as
  - Domain adaptation: MBV3-large/small (labeled source) → MBV3-small (non-labeled target) || Technique examples: Reweighting, DANN, CORAL, MMD/DDC, or newer.
    Ref: https://github.com/thuml/Transfer-Learning-Library/tree/master/tllib
- Model compression [W6-W7], such as
  - Knowledge distillation (KD): MBV3-large/small (labeled source) → MBV3-small (non-labeled target)
  - Pruning (normal, iterative, etc.) on MBV3-small (non-labeled target)
  - Quantization (PTDQ, PTSQ, QAT, etc.) on MBV3-small (non-labeled target)
  - Low-rank approximation (SVD, Higher-order SVD [CP, Tucker], etc.), or newer, on MBV3-small (non-labeled target)
    Ref: https://github.com/juliagusak/model-compression-and-acceleration-progress/blob/master/README.md
- Reduce model precision [W7], i.e., applying XNOR convolution layer, mixed precision, etc.
- Optimizing runtime [W7], i.e., through ONNXruntime/TensorRT library or any other packages

Dataset for scoring: ***Office-31*** [3 domains (~2,9xx images): *Amazon* (A), *Webcam* (W), *DSLR* (D)] (Reference: https://github.com/jindongwang/transferlearning/blob/master/data/dataset.md)

The number of iterations of the training source dataset: As you desired (If any pretraining)

The number of iterations of the training unlabeled target dataset for scoring: 30-50 rounds

The number of iterations of the training labeled target (baseline): Equal to unlabeled target task above

Table I. Scoring criteria

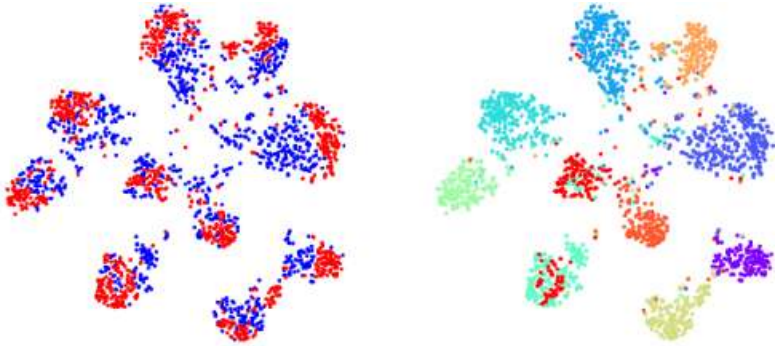
Evaluation topics	The performance is excellent, close to *MBV3-s* as per the comparative description below [From the lesson learned]	Scoring if good close to *MBV3-s (labeled target)*	Scoring if just only finished
Basic base			
Quantitative benchmarks	Accuracy, loss, precision, recall, F1-score, confusion matrix, NMI-RI score, etc. [W3-W4]	9-10 %	4-5 %
Resource usage	Number of parameters, computational power (FLOPs), etc. [W3-W7]		
Visualization	Plots of performance (accuracy & loss) from both training & validation rounds [W2-W3], t-SNE, A-distance [W5], etc.		
Acceleration	Higher FPS/frame rate [W7] - Require: FP16 || Select: FP8 or INT8 - Tested on normal instances (Colab/others)	4.5-5 %	2-2.5 %
Source code	Source code in use for benchmarking: Optimizer, loss_fn, improvement techniques, etc. [Every week that passed]	4.5-5 %	2-2.5 %
Report	Summarize everything from your projects (at least 2-5 pages)		
Presentation	Presentation (just 5-15 mins) and demo recorded (submitting post-presentations)		
Extra base			
Extra points 1: **The best of performance**	Anyone who got the best performance closing at *MobileNetV3-small* (*MBV3-s*) for above overall benchmarks	Top-1: +3 % Top-2: +2 % Top-3: +1 %	-
Extra points 2: **The lightest weight of the model**	Anyone who can optimize *MobileNetV3-small* (*MBV3-s*) become the lowest resource consumption, e.g., number of parameters (units) ↓, reduce model size (MB) ↓, reduce training-/inference time consumption (sec) ↓, etc.	Top-1: +3 % Top-2: +2 % Top-3: +1 %	-
Total		18-20 % + (1-6 % extra)	8-10 %

Appendix A. Example of benchmarking performance (%acc) in transfer task of domain1 at *MBV3-l* → domain2 at *MBV3-s*

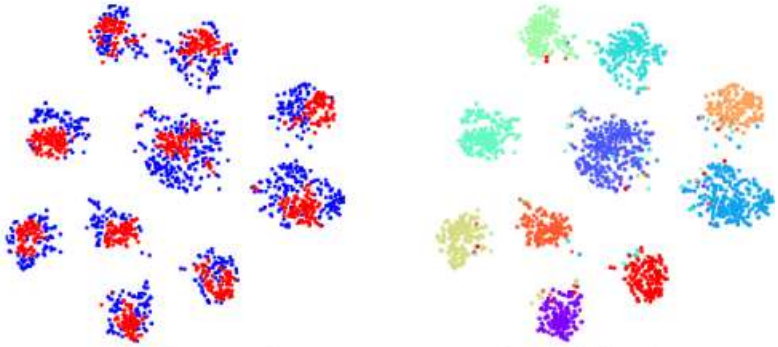
Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
AlexNet [50]	45.2	88.1	96.2	47.4	38.2	36.1	58.5
TCA [17]	61.0	93.2	95.2	60.8	51.6	50.9	68.8
GFK [51]	60.4	95.6	95.0	60.6	52.4	48.1	68.7
D-CORAL [4]	61.6	95.7	99.2	66.8	52.8	51.5	72.1
DAN [48]	68.5	96.0	99.0	67.0	54.0	53.1	72.9
DANN [34]	73.0	96.4	99.2	72.3	53.4	51.2	74.3
ADDA [7]	73.5	96.2	98.8	71.6	54.6	53.5	74.7
JAN [12]	74.9	96.6	99.5	71.8	58.3	55.0	76.0
CDAN [52]	**77.9**	96.9	**100.0**	74.6	55.1	57.5	77.0
ResNet-50 [13]	68.3	96.7	99.2	69.1	61.7	60.0	75.8
MobileDA	71.5	**97.4**	99.8	**75.3**	**63.4**	**62.1**	**78.3**

Remark Benchmark 3 results: 1. **Source-Only** (i.e., train source, predict target), 2. **your techniques**, and 3. **full-labeled target training (Baseline)** (i.e., train target, predict target)

Appendix B. t-SNE examples



(a) DANN (by domain) (b) DANN (by classes)



(c) ALIC (by domain) (d) ALIC (by classes)

FIGURE 4. Feature visualization for embedding of digit datasets for adapting SVHN to MNIST using t-SNE algorithm. Source and target samples are denoted as blue and red points in the first column. Each

Appendix C. MobileNetV3-small model

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d, 3x3	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	✓	RE	2
$56^2 \times 16$	bneck, 3x3	72	24	-	RE	2
$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	✓	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	✓	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	144	48	✓	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	✓	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	✓	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	✓	HS	1
$7^2 \times 576$	pool, 7x7	-	-	-	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024	-	HS	1
$1^2 \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Table 2. Specification for MobileNetV3-Small. See table 1 for notation.

Appendix D. MobileNetV3-large model

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Table 1. Specification for MobileNetV3-Large. SE denotes whether there is a Squeeze-And-Excite in that block. NL denotes the type of nonlinearity used. Here, HS denotes h-swish and RE denotes ReLU. NBN denotes no batch normalization. s denotes stride.

Appendix E. Office-31 Dataset Example (three-domain 31 classes)