MobilenetV2 and above

For MobilenetV2+ see this file mobilenet/README.md

MobileNetV1

MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings and segmentation similar to how other popular large scale models, such as Inception, are used. MobileNets can be run efficiently on mobile devices with TensorFlow Lite.

MobileNets trade off between latency, size and accuracy while comparing favorably with popular models from the literature.

alt text

Pre-trained Models

Choose the right MobileNet model to fit your latency and size budget. The size of the network in memory and on disk is proportional to the number of parameters. The latency and power usage of the network scales with the number of Multiply-Accumulates (MACs) which measures the number of fused Multiplication and Addition operations. These MobileNet models have been trained on the ILSVRC-2012-CLS image classification dataset. Accuracies were computed by evaluating using a single image crop.

Model	Million MACs	Million Parameters	Top-1 Accuracy	Top-5 Accuracy
MobileNet_v1_1.0_224	569	4.24	70.9	89.9
MobileNet_v1_1.0_192	418	4.24	70.0	89.2
MobileNet_v1_1.0_160	291	4.24	68.0	87.7
MobileNet_v1_1.0_128	186	4.24	65.2	85.8
MobileNet_v1_0.75_224	317	2.59	68.4	88.2
MobileNet_v1_0.75_192	233	2.59	67.2	87.3
MobileNet_v1_0.75_160	162	2.59	65.3	86.0
MobileNet_v1_0.75_128	104	2.59	62.1	83.9
MobileNet_v1_0.50_224	150	1.34	63.3	84.9
MobileNet_v1_0.50_192	110	1.34	61.7	83.6
MobileNet_v1_0.50_160	77	1.34	59.1	81.9
MobileNet_v1_0.50_128	49	1.34	56.3	79.4
MobileNet_v1_0.25_224	41	0.47	49.8	74.2
MobileNet_v1_0.25_192	34	0.47	47.7	72.3
MobileNet_v1_0.25_160	21	0.47	45.5	70.3
MobileNet_v1_0.25_128	14	0.47	41.5	66.3

Model	Million MACs	Million Parameters	Top-1 Accuracy	Top-5 Accuracy
MobileNet_v1_1.0_224_quant	569	4.24	70.1	88.9
MobileNet_v1_1.0_192_quant	418	4.24	69.2	88.3
MobileNet_v1_1.0_160_quant	291	4.24	67.2	86.7
MobileNet_v1_1.0_128_quant	186	4.24	63.4	84.2
MobileNet_v1_0.75_224_quant	317	2.59	66.8	87.0
MobileNet_v1_0.75_192_quant	233	2.59	66.1	86.4
MobileNet_v1_0.75_160_quant	162	2.59	62.3	83.8
MobileNet_v1_0.75_128_quant	104	2.59	55.8	78.8
MobileNet_v1_0.50_224_quant	150	1.34	60.7	83.2
MobileNet_v1_0.50_192_quant	110	1.34	60.0	82.2
MobileNet_v1_0.50_160_quant	77	1.34	57.7	80.4
MobileNet_v1_0.50_128_quant	49	1.34	54.5	77.7
MobileNet_v1_0.25_224_quant	41	0.47	48.0	72.8
MobileNet_v1_0.25_192_quant	34	0.47	46.0	71.2
MobileNet_v1_0.25_160_quant	21	0.47	43.4	68.5
MobileNet_v1_0.25_128_quant	14	0.47	39.5	64.4

Revisions to models: * July 12, 2018: Update to TFLite models that fixes an accuracy issue resolved by making conversion support weights with narrow_range. We now report validation on the actual TensorFlow Lite model rather than the emulated quantization number of TensorFlow. * August 2, 2018: Update to TFLite models that fixes an accuracy issue resolved by making sure the numerics of quantization match TF quantized training accurately.

The linked model tar files contain the following: * Trained model checkpoints * Eval graph text protos (to be easily viewed) * Frozen trained models * Info file containing input and output information * Converted TensorFlow Lite flatbuffer model

Note that quantized model GraphDefs are still float models, they just have FakeQuantization operation embedded to simulate quantization. These are converted by TensorFlow Lite to be fully quantized. The final effect of quantization can be seen by comparing the frozen fake quantized graph to the size of the TFLite flatbuffer, i.e. The TFLite flatbuffer is about 1/4 the size. For more information on the quantization techniques used here, see here. There isn't any equivalent in TF2.x yet, more information can be found in this RFC

Here is an example of how to download the MobileNet_v1_1.0_224 checkpoint:

```
$ CHECKPOINT_DIR=/tmp/checkpoints
$ mkdir ${CHECKPOINT_DIR}
$ wget http://download.tensorflow.org/models/mobilenet_v1_2018_02_22/mobilenet_v1_1.0_224.tg
$ tar -xvf mobilenet_v1_1.0_224.tgz
$ mv mobilenet_v1_1.0_224.ckpt.* ${CHECKPOINT_DIR}
```

MobileNet V1 scripts

This package contains scripts for training floating point and eight-bit fixed point TensorFlow models.

Quantization tools used are described here. There isn't any equivalent in TF2.x yet, more information can be found in this RFC

Conversion to fully quantized models for mobile can be done through TensorFlow Lite.

Usage

Build for GPU

```
$ bazel build -c opt --config=cuda mobilenet_v1_{eval,train}
```

Running

Float Training and Eval Train:

```
$ ./bazel-bin/mobilenet_v1_train --dataset_dir "path/to/dataset" --checkpoint_dir "path/to/c
```

Eval:

```
$ ./bazel-bin/mobilenet_v1_eval --dataset_dir "path/to/dataset" --checkpoint_dir "path/to/ch
```

Quantized Training and Eval Train from preexisting float checkpoint:

```
$ ./bazel-bin/mobilenet_v1_train --dataset_dir "path/to/dataset" --checkpoint_dir "path/to/c
  --quantize=True --fine_tune_checkpoint=float/checkpoint/path
```

Train from scratch:

```
$ ./bazel-bin/mobilenet_v1_train --dataset_dir "path/to/dataset" --checkpoint_dir "path/to/c
```

Eval:

```
$ ./bazel-bin/mobilenet_v1_eval --dataset_dir "path/to/dataset" --checkpoint_dir "path/to/ch
```

The resulting float and quantized models can be run on-device via TensorFlow Lite.