# TFLite Converter segfaults when trying to convert per-channel quantized transposed convolutions #53767

Open lgeiger opened this issue on Jan 14 · 5 comments

Assignees 

Labels stat:awaiting tensorflower TF 2.7 TFLiteConverter type:bug

lgeiger commented on Jan 14 • edited Contributor

When converting transposed convolutions using per-channel weight quantization the converter segfaults and crashes the Python process. Per-channel quantization is supported by TFLite Transposed convolutions:

tensorflow/tensorflow/lite/kernels/transpose_conv.cc
Lines 371 to 380 in f87be6c

```
371     TF_LITE_ENSURE_EQ(context, weights->quantization.type,
372                       kTfLiteAffineQuantization);
373     const auto* affine_quantization =
374         reinterpret_cast<TfLiteAffineQuantization*>(
375             weights->quantization.params);
376     const int channels_out = weights->dims->data[0];
377     TF_LITE_ENSURE(context, affine_quantization);
378     TF_LITE_ENSURE(context, affine_quantization->scale);
379     TF_LITE_ENSURE(context, (affine_quantization->scale->size == 1 ||
380                              affine_quantization->scale->size == channels_out));
```

so the converter shouldn't segfault when trying to convert such a model.

It looks like this issue has been introduced in TensorFlow 2.6 since the same model code produced a valid TFLite file in TensorFlow 2.5. This issue might also be related to #53766, but in any case the converter should never segfault.

## 1. System information

- OS Platform and Distribution (e.g., Linux Ubuntu 16.04): macOS / Ubuntu

- TensorFlow installation (pip package or built from source): pip package
- TensorFlow library (version, if pip package or github SHA, if built from source): 2.6, 2.7, 2.8rc0 and 2.9.0-dev20220114

## 2. Code

A minimal reproduction of the issue and a workaround is available in this notebook.

```
import tensorflow as tf


class QuantConv2DTransposed(tf.keras.layers.Layer):
    def build(self, input_shape):
        self.kernel = self.add_weight("kernel", [3, 3, input_shape[-1], 24])

    def call(self, inputs):
        filters = tf.quantization.fake_quant_with_min_max_vars_per_channel(
            self.kernel, -3.0 * tf.ones([24]), 3.0 * tf.ones([24]), narrow_range=True
        )
        filters = tf.transpose(filters, (0, 1, 3, 2))
        return tf.nn.conv2d_transpose(inputs, filters, [*inputs.shape[:-1], 24], 1)


inp = tf.keras.Input(shape=(6, 8, 48), batch_size=1)
x = tf.quantization.fake_quant_with_min_max_vars(inp, -3.0, 3.0, narrow_range=True)
x = QuantConv2DTransposed()(x)
x = tf.quantization.fake_quant_with_min_max_vars(x, -3.0, 3.0, narrow_range=True)

model = tf.keras.Model(inp, x)

model.save("/tmp/testing")
converter = tf.lite.TFLiteConverter.from_saved_model("/tmp/testing")
converter.optimizations = [tf.lite.Optimize.DEFAULT]

# terminated by signal SIGSEGV (Address boundary error)
tflite_model = converter.convert()
```

👍 1

lgeiger added the TFLiteConverter label on Jan 14

google-ml-butler bot assigned sushreebarsa on Jan 14

tilakrayal mentioned this issue on Jan 17

**Constant folding fails when converting int8 transposed convolutions** #53766

Closed

sushreebarsa added TF 2.7 type:bug labels on Jan 17

**sushreebarsa** commented on Jan 17 — Contributor

@Saduf2019 Was able to replicate the issue on colab using TF v2.6.0, 2.7.0 and tf-nightly(2.9.0-dev20220114),please find the attached gists.Thank you!

sushreebarsa assigned Saduf2019 and unassigned sushreebarsa on Jan 17

**lgeiger** commented on Jan 17 — Contributor, Author

> Was able to replicate the issue on colab using TF v2.6.0, 2.7.0 and tf-nightly(2.9.0-dev20220114),please find the attached gists

@sushreebarsa Thanks for confirming. Just for reference, your reproduction on TF 2.6.0 actually now fails due to an unrelated Keras version conflict. Changing the dependency from `v2.6.0` to `v2.6.2` will fix this and allow you to correctly reproduce the segfault mentioned in this issue.

👍 1

Saduf2019 assigned jvishnuvardhan and unassigned Saduf2019 on Jan 17

jvishnuvardhan assigned JunyoungLim and unassigned jvishnuvardhan on Jan 20

jvishnuvardhan added the stat:awaiting tensorflower label on Jan 20

**lgeiger** commented on Mar 18 — Contributor, Author

This is still an issue in `2.9.0-dev20220318`. Are there any updates on this?
Being able to trigger a converter segfault seems to be quite problematic.

**lgeiger** commented on May 11 — Contributor, Author

@JunyoungLim I retested the above example `2.10.0-dev20220427` and the converter still segfaults.

**lgeiger** commented on Sep 13 Contributor Author

I retested the above example with `2.10.0` and the segfault seems to be fixed now, however conversion still fails with:

```
'tfl.transpose' op has mismatched quantized axes of input and output
```

See here.

Assignees

JunyoungLim

Labels

stat:awaiting tensorflower TF 2.7 TFLiteConverter type:bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

5 participants