Module: tf.contrib.deprecated

Defined in `tensorflow/contrib/deprecated/__init__.py`.

Non-core alias for the deprecated tf.X_summary ops.

For TensorFlow 1.0, we have reorganized the TensorFlow summary ops into a submodule, and made some semantic tweaks. The first thing to note is that we moved the APIs around as follows:

```
tf.scalar_summary -> tf.summary.scalar
tf.histogram_summary -> tf.summary.histogram
tf.audio_summary -> tf.summary.audio
tf.image_summary -> tf.summary.image
tf.merge_summary -> tf.summary.merge
tf.merge_all_summaries -> tf.summary.merge_all
```

We think this API is cleaner and will improve long-term discoverability and clarity of the TensorFlow API. But we also took the opportunity to make an important change to how summary "tags" work. The "tag" of a summary is the string that is associated with the output data, i.e. the key for organizing the generated protobufs.

Previously, the tag was allowed to be any unique string; it had no relation to the summary op generating it, and no relation to the TensorFlow name system. This behavior made it very difficult to write reusable that would add summary ops to the graph. If you had a function to add summary ops, you would need to pass in a `tf.name_scope`, manually, to that function to create deduplicated tags. Otherwise your program would fail with a runtime error due to tag collision.

The new summary APIs under `tf.summary` throw away the "tag" as an independent concept; instead, the first argument is the node name. So summary tags now automatically inherit the surrounding `tf.name_scope`, and automatically are deduplicated if there is a conflict. Now however, the only allowed characters are alphanumerics, underscores, and forward slashes. To make migration easier, the new APIs automatically convert illegal characters to underscores.

Just as an example, consider the following "before" and "after" code snippets:

```
# Before
def add_activation_summaries(v, scope):
  tf.scalar_summary("%s/fraction_of_zero" % scope, tf.nn.fraction_of_zero(v))
  tf.histogram_summary("%s/activations" % scope, v)

# After
def add_activation_summaries(v):
  tf.summary.scalar("fraction_of_zero", tf.nn.fraction_of_zero(v))
  tf.summary.histogram("activations", v)
```

Now, so long as the add_activation_summaries function is called from within the right `tf.name_scope`, the behavior is the same.

Because this change does modify the behavior and could break tests, we can't automatically migrate usage to the new APIs. That is why we are making the old APIs temporarily available here at `tf.contrib.deprecated`.

In addition to the name change described above, there are two further changes to the new summary ops:

- the "max_images" argument for `tf.image_summary` was renamed to "max_outputs for `tf.summary.image`

- `tf.scalar_summary` accepted arbitrary tensors of tags and values. But `tf.summary.scalar` requires a single scalar name and scalar value. In most cases, you can create `tf.summary.scalar` in a loop to get the same behavior

As before, TensorBoard groups charts by the top-level `tf.name_scope` which may be inconvenient, for in the new summary ops, the summary will inherit that `tf.name_scope` without user control. We plan to add more grouping mechanisms to TensorBoard, so it will be possible to specify the TensorBoard group for each summary via the summary API.

Functions

`audio_summary(...)` : Outputs a `Summary` protocol buffer with audio. (deprecated)

`histogram_summary(...)` : Outputs a `Summary` protocol buffer with a histogram. (deprecated)

`image_summary(...)` : Outputs a `Summary` protocol buffer with images. (deprecated)

`merge_all_summaries(...)` : Merges all summaries collected in the default graph. (deprecated)

`merge_summary(...)` : Merges summaries. (deprecated)

`scalar_summary(...)` : Outputs a `Summary` protocol buffer with scalar values. (deprecated)

Last updated November 2, 2017.