

过渡到TensorFlow 1.0

- 如何升级
 - 限制
- 手动升级代码
 - 变量
 - 总结功能
 - 数值差异
 - NumPy匹配名称
 - NumPy匹配参数
 - 简化的数学变体
 - 其他更改

原文链接：<https://www.tensorflow.org/install/migration>

译文链接：<http://www.apache.wiki/pages/viewpage.action?pageId=10029601>

贡献者：片刻 ApacheCN Apache中文网

TensorFlow 1.0中的API已经改变了不是全部向后兼容的方式。也就是说，使用TensorFlow 0.n的TensorFlow程序不一定适用于TensorFlow 1.0。我们已经对此API进行了更改，以确保内部一致的API，并且不打算在整个1.N生命周期内做出反向变化。

本指南将引导您了解API的主要更改以及如何自动升级TensorFlow 1.0的程序。本指南不仅可以帮助您完成更改，还可以解释为什么我们已经做出了这些更改。

如何升级

如果你想自动将你的代码移植到1.0，你可以尝试我们的 `tf_upgrade.py` 脚本。虽然此脚本处理许多情况，但有时需要手动更改。从我们的GitHub树中获取这个脚本。

要将单个0.n TensorFlow源文件转换为1.0，请输入以下格式的命令：

```
$ python tf_upgrade.py --infile InputFile --outfile OutputFile
```

test.py“1.0 TensorFlow” 0.n TensorFlowtest_1.0.py

```
$ python tf_upgrade.py --infile test.py --outfile test_1.0.py
```

tf_upgrade.pyreport.txt

要将0.n TensorFlow程序的整个目录升级到1.0，请输入以下格式的命令：

```
$ python tf_upgrade.py --intree InputDir --outtree OutputDir
```

0.n TensorFlow/home/user/cool1.0/home/user/cool_1.0

```
$ python tf_upgrade.py --intree /home/user/cool --outtree /home/user/cool_1.0
```

限制

有几件事要注意。特别：

- 您必须手动修复任何实例`tf.reverse()`。该`tf_upgrade.py`脚本将发出警告，`tf.reverse()`在标准输出和在`report.txt`文件中。
- 在重新排序的参数中，`tf_upgrade.py`尝试最小化重新格式化代码，因此不能自动更改实际的参数顺序。相反，`tf_upgrade.py`通过引入关键字参数，使您的函数调用与订单无关。
- 这样的建筑`tf.get_variable_scope().reuse_variables()`可能不会奏效。我们建议删除这些行并用以下行替换它们：

```
with tf.variable_scope(tf.get_variable_scope(), reuse=True):
    ...
```

- 类似于`tf.pack`并且`tf.unpack`，我们正在重新命名`TensorArray.pack`，并`TensorArray.unpack`以`TensorArray.stack`和`TensorArray.unstack`。然而，`TensorArray.pack` 并且`TensorArray.unpack`不能词法，因为它们是间接相关的检测到的`tf`命名空间如`foo = tf.TensorArray(); foo.unpack()`

手动升级代码

`tf_upgrade.py`您可以手动升级代码而不是运行。本文档的其余部分提供了TensorFlow 1.0中所有向后不兼容更改的全面列表。

变量

可变函数已经变得更加一致，更不容易混淆。

- `tf.VARIABLES`
 - 应该改名 `tf.GLOBAL_VARIABLES`
- `tf.all_variables`
 - 应该改名 `tf.global_variables`
- `tf.initialize_all_variables`
 - 应该改名 `tf.global_variables_initializer`
- `tf.initialize_local_variables`
 - 应该改名 `tf.local_variables_initializer`
- `tf.initialize_variables`
 - 应该改名 `tf.variables_initializer`

总结功能

摘要功能已在命名空间下`tf.summary`合并。

- `tf.audio_summary`
 - 应该改名 `tf.summary.audio`
- `tf.contrib.deprecated.histogram_summary`
 - 应该改名 `tf.summary.histogram`
- `tf.contrib.deprecated.scalar_summary`
 - 应该改名 `tf.summary.scalar`
- `tf.histogram_summary`
 - 应该改名 `tf.summary.histogram`
- `tf.image_summary`
 - 应该改名 `tf.summary.image`
- `tf.merge_all_summaries`
 - 应该改名 `tf.summary.merge_all`
- `tf.merge_summary`
 - 应该改名 `tf.summary.merge`
- `tf.scalar_summary`
 - 应该改名 `tf.summary.scalar`
- `tf.train.SummaryWriter`
 - 应该改名 `tf.summary.FileWriter`

数值差异

整数分区，`tf.floordiv`现在使用地板语义。这是为了使结果`np.divide`和`np.mod`一致，`tf.divide`并`tf.mod`分别。另外我们改变了用于`tf.round`匹配NumPy的舍入算法。

- `tf.div`
 - `tf.divide`划分的语义已经完全改变为符合Python语义。也就是说，//在Python 3中，Python 2中的未来分区模式将始终产生浮点数，//将产生分区。然而，甚至`tf.div`会产生地板整数除法。要强制使用C风格的截断语义，你必须使用`tf.truncatediv`。

- 考虑改变你要使用的代码`tf.divide`，它遵循Python语义进行升级。
- `tf.mod`
 - 的语义`tf.mod`已更改为符合Python的语义。特别地，地板语义用于整数。如果您想要使用C型截断模式（余数），可以使用`tf.truncatemod`

可以用这个表来总结分裂的新旧行为：

EXPR	TF 0.11 (Py2)	TF 0.11 (py3)	TF 1.0 (py2)	TF 1.0 (py3)
<code>tf.div (3,4)</code>	0	0	0	0
<code>tf.div (-3,4)</code>	0	0	-1	-1
<code>tf.mod (-3,4)</code>	-3	-3	1	1
<code>-3/4</code>	0	-0.75	-1	-0.75
<code>-3 / 4tf.divide (-3,4)</code>	N / A	N / A	-0.75	-1

四舍五入的新旧行为可以概括如下：

输入	Python	NumPy	C ++ round ()	TensorFlow 0.11 (floor (x + .5))	TensorFlow 1.0
-3.5	-4	-4	-4	-3	-4
-2.5	-2	-2	-3	-2	-2
-1.5	-2	-2	-2	-1	-2
-0.5	0	0	-1	0	0
0.5	0	0	1	1	0
1.5	2	2	2	2	2
2.5	2	2	3	3	2
3.5	4	4	4	4	4

NumPy匹配名称

许多功能已重命名为匹配NumPy。这样做是为了使NumPy和TensorFlow之间的转换尽可能简单。还有许多功能不匹配的情况，所以这远远不是一个艰难和快速的规则，但是我们已经删除了几个常见的不一致。

- `tf.inv`
 - 应该改名 `tf.reciprocal`
 - 这样做是为了避免与NumPy的矩阵逆相混淆 `np.inv`
- `tf.list_diff`
 - 应该改名 `tf.setdiff1d`
- `tf.listdiff`
 - 应该改名 `tf.setdiff1d`
- `tf.mul`
 - 应该改名 `tf.multiply`
- `tf.neg`
 - 应该改名 `tf.negative`
- `tf.select`
 - 应该改名 `tf.where`
 - `tf.where` 现在有3个参数或1个参数，就像 `np.where`
- `tf.sub`
 - 应该改名 `tf.subtract`

NumPy匹配参数

某些TensorFlow 1.0方法的参数现在匹配某些NumPy方法中的参数。为了实现这一点，TensorFlow 1.0已经改变了关键字参数并重新排列了一些参数。值得注意的是，TensorFlow 1.0现在使用`axis`而不是`dimension`。TensorFlow 1.0旨在保持张量参数首先修改Tensors的操作。（见`tf.concat`更改）。

- `tf.argmax`
 - 关键字参数`dimension`应该重命名为`axis`
- `tf.argmin`
 - 关键字参数`dimension`应该重命名为`axis`
- `tf.concat`

- 关键字参数concat_dim应该重命名为axis
- 论据已被重新排列tf.concat(values, axis, name='concat')。
- tf.count_nonzero
 - 关键字参数reduction_indices应该重命名为axis
- tf.expand_dims
 - 关键字参数dim应该重命名为axis
- tf.reduce_all
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_any
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_join
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_logsumexp
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_max
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_mean
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_min
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_prod
 - 关键字参数reduction_indices应该重命名为axis
- tf.reduce_sum
 - 关键字参数reduction_indices应该重命名为axis
- tf.reverse
 - tf.reverse曾经采取1D bool张量来控制哪些尺寸被扭转。现在我们使用轴索引的张量。
 - 例如tf.reverse(a, [True, False, True])现在一定是tf.reverse(a, [0, 2])
- tf.reverse_sequence
 - 关键字参数batch_dim应该重命名为batch_axis
 - 关键字参数seq_dim应该重命名为seq_axis
- tf.sparse_concat
 - 关键字参数concat_dim应该重命名为axis
- tf.sparse_reduce_sum
 - 关键字参数reduction_axes应该重命名为axis
- tf.sparse_reduce_sum_sparse
 - 关键字参数reduction_axes应该重命名为axis
- tf.sparse_split
 - 关键字参数split_dim应该重命名为axis
 - 论据已被重新排列tf.sparse_split(keyword_required=KeywordRequired(), sp_input=None, num_split=None, axis=None, name=None, split_dim=None)。
- tf.split
 - 关键字参数split_dim应该重命名为axis
 - 关键字参数num_split应该重命名为num_or_size_splits
 - 论据已被重新排列tf.split(value, num_or_size_splits, axis=0, num=None, name='split')。
- tf.squeeze
 - 关键字参数squeeze_dims应该重命名为axis
- tf.svd
 - 论据已被重新排列tf.svd(tensor, full_matrices=False, compute_uv=True, name=None)。

简化的数学变体

批量版数学运算已被删除。现在，功能包含在非批量版本中。同样，tf.complex_abs已经将其功能转移到了tf.abs

- tf.batch_band_part
 - 应该改名 tf.band_part
- tf.batch_cholesky
 - 应该改名 tf.cholesky
- tf.batch_cholesky_solve
 - 应该改名 tf.cholesky_solve
- tf.batch_fft
 - 应该改名 tf.fft
- tf.batch_fft3d
 - 应该改名 tf.fft3d
- tf.batch_ifft
 - 应该改名 tf.ifft
- tf.batch_ifft2d
 - 应该改名 tf.ifft2d

- `tf.batch_ifft3d`
 - 应该改名 `tf.ifft3d`
- `tf.batch_matmul`
 - 应该改名 `tf.matmul`
- `tf.batch_matrix_determinant`
 - 应该改名 `tf.matrix_determinant`
- `tf.batch_matrix_diag`
 - 应该改名 `tf.matrix_diag`
- `tf.batch_matrix_inverse`
 - 应该改名 `tf.matrix_inverse`
- `tf.batch_matrix_solve`
 - 应该改名 `tf.matrix_solve`
- `tf.batch_matrix_solve_ls`
 - 应该改名 `tf.matrix_solve_ls`
- `tf.batch_matrix_transpose`
 - 应该改名 `tf.matrix_transpose`
- `tf.batch_matrix_triangular_solve`
 - 应该改名 `tf.matrix_triangular_solve`
- `tf.batch_self_adjoint_eig`
 - 应该改名 `tf.self_adjoint_eig`
- `tf.batch_self_adjoint_eigvals`
 - 应该改名 `tf.self_adjoint_eigvals`
- `tf.batch_set_diag`
 - 应该改名 `tf.set_diag`
- `tf.batch_svd`
 - 应该改名 `tf.svd`
- `tf.complex_abs`
 - 应该改名 `tf.abs`

其他更改

已经做了几个其他变化，其中包括：

- `tf.image.per_image_whitening`
 - 应该改名 `tf.image.per_image_standardization`
- `tf.nn.sigmoid_cross_entropy_with_logits`
 - 论据已被重新排列`tf.nn.sigmoid_cross_entropy_with_logits(_sentinel=None, labels=None, logits=None, name=None)`。
- `tf.nn.softmax_cross_entropy_with_logits`
 - 论据已被重新排列`tf.nn.softmax_cross_entropy_with_logits(_sentinel=None, labels=None, logits=None, dim=-1, name=None)`。
- `tf.nn.sparse_softmax_cross_entropy_with_logits`
 - 论据已被重新排列`tf.nn.sparse_softmax_cross_entropy_with_logits(_sentinel=None, labels=None, logits=None, name=None)`。
- `tf.ones_initializer`
 - 应该改为函数调用`tf.ones_initializer()`
- `tf.pack`
 - 应该改名 `tf.stack`
- `tf.round`
 - `tf.round`现在的语义符合银行家的四舍五入。
- `tf.unpack`
 - 应该改名 `tf.unstack`
- `tf.zeros_initializer`
 - 应该改为函数调用`tf.zeros_initializer()`