

Rock, Paper, Scissors

In this week's exercise you will be working with TFDS and the rock-paper-scissors dataset. You'll do a few tasks such as exploring the info of the dataset in order to figure out the name of the splits. You'll also write code to see if the dataset supports the new S3 API before creating your own versions of the dataset.

Setup

In [3]:

```
# Use all imports
from os import getcwd

import tensorflow as tf
import tensorflow_datasets as tfds
```

Extract the Rock, Paper, Scissors Dataset

In the cell below, you will extract the `rock_paper_scissors` dataset and then print its info. Take note of the splits, what they're called, and their size.

In [8]:

```
# EXERCISE: Use tfds.load to extract the rock_paper_scissors dataset.
```

```
filePath = f"{getcwd()}/../tmp2"
data, info = tfds.load(name='rock_paper_scissors', data_dir=filePath, with_info=True)
print(info)
```

WARNING:absl:Found a different version 3.0.0 of dataset rock_paper_scissors in data_dir /tf/week1/../tmp2. Using currently defined version 1.0.0.

```
tfds.core.DatasetInfo(
  name='rock_paper_scissors',
  version=1.0.0,
  description='Images of hands playing rock, paper, scissor game.',
  urls=['http://laurencemoroney.com/rock-paper-scissors-dataset'],
  features=FeaturesDict({
    'image': Image(shape=(300, 300, 3), dtype=tf.uint8),
    'label': ClassLabel(shape=(), dtype=tf.int64, num_classes=3),
  }),
  total_num_examples=2892,
  splits={
    'test': 372,
    'train': 2520,
  },
  supervised_keys=('image', 'label'),
  citation="""@ONLINE {rps,
author = "Laurence Moroney",
title = "Rock, Paper, Scissors Dataset",
month = "feb",
year = "2019",
url = "http://laurencemoroney.com/rock-paper-scissors-dataset"
}""",
  redistribution_info=,
)
```

In [10]:

```
# EXERCISE: In the space below, write code that iterates through the splits
# without hardcoding any keys. The code should extract 'test' and 'train' as
# the keys, and then print out the number of items in the dataset for each key.
# HINT: num_examples property is very useful here.
```

```
for k, v in data.items():
    print("{}:{}".format(k, info.splits[k].num_examples))

# EXPECTED OUTPUT
# test:372
# train:2520
```

```
test:372
train:2520
```

Use the New S3 API

Before using the new S3 API, you must first find out whether the `rock_paper_scissors` dataset implements the new S3 API. In the cell below you should use version `3.*.*` of the `rock_paper_scissors` dataset.

In [11]:

```
# EXERCISE: In the space below, use the tfds.builder to fetch the
# rock_paper_scissors dataset and check to see if it supports the
# new S3 API.
# HINT: The builder should 'implement' something

rps_builder = tfds.builder("rock_paper_scissors:3.*.*", data_dir=filePath)

print(rps_builder.version.implements(tfds.core.Experiment.S3))

# Expected output:
# True
```

```
True
```

Create New Datasets with the S3 API

Sometimes datasets are too big for prototyping. In the cell below, you will create a smaller dataset, where instead of using all of the training data and all of the test data, you instead have a `small_train` and `small_test` each of which are comprised of the first 10% of the records in their respective datasets.

In [12]:

```
# EXERCISE: In the space below, create two small datasets, `small_train`
# and `small_test`, each of which are comprised of the first 10% of the
# records in their respective datasets.

small_train = tfds.load("rock_paper_scissors:3.*.*", data_dir=filePath, split='train[:10%]')
small_test = tfds.load("rock_paper_scissors:3.*.*", data_dir=filePath, split='test[:10%]')

# No expected output yet, that's in the next cell
```

In [13]:

```
# EXERCISE: Print out the size (length) of the small versions of the datasets.

print(len(list(small_train)))
print(len(list(small_test)))

# Expected output
# 252
# 37
```

```
252
37
```

The original dataset doesn't have a validation set, just training and testing sets. In the cell below, you will use TFDS to create new datasets according to these rules:

- `new_train`: The new training set should be the first 90% of the original training set.

- `new_test` : The new test set should be the first 90% of the original test set.
- `validation` : The new validation set should be the last 10% of the original training set + the last 10% of the original test set.

In [15]:

```
# EXERCISE: In the space below, create 3 new datasets according to
# the rules indicated above.

new_train = tfds.load("rock_paper_scissors:3.*.*", data_dir=filePath, split='train[:90%]')
print(len(list(new_train)))

new_test = tfds.load("rock_paper_scissors:3.*.*", data_dir=filePath, split='test[:90%]')
print(len(list(new_test)))

validation = tfds.load("rock_paper_scissors:3.*.*", data_dir=filePath, split='train[90%:] +
test[90%:]')
print(len(list(validation)))

# Expected output
# 2268
# 335
# 289
```

```
2268
335
289
```

Submission Instructions

In []:

```
# Now click the 'Submit Assignment' button above.
```

When you're done or would like to take a break, please run the two cells below to save your work and close the Notebook. This frees up resources for your fellow learners.

In []:

```
%%javascript
<!-- Save the notebook -->
IPython.notebook.save_checkpoint();
```

In []:

```
%%javascript
<!-- Shutdown and close the notebook -->
window.onbeforeunload = null
window.close();
IPython.notebook.session.delete();
```