

Assignment 1 (key)

1. Given the following dataset, address each part by creating code and executing code in the Colab environment with TensorFlow (show code and output).

[[0, 1, 9, 8, 7, 3], [2, 9, 4, 0, 2, 3], [7, 3, 3, 2, 2, 1], [0, 0, 1, 2, 2, 5]]

- a. create a TensorFlow dataset with the **from_tensor_slices** method:
- b. iterate and display elements as tensors **and** as NumPy arrays:
- c. display the 3rd element from the 2nd tensor:

(hint: convert dataset to iterator with **iter** method)

2. Enable the GPU available from Colab (show code and output):
3. Download data with **tf.keras.utils.get_file** utility into pandas dataframe.

- a. use the following URL:

<https://archive.ics.uci.edu/ml/machine-learning-databases/flags/flag.data>

- b. use the following dataset: **flag.data**

- c. use the utility to get the appropriate dataset path.

```
# we need the keras module
from tensorflow import keras

ds = 'flag.data'
piece1 = 'https://archive.ics.uci.edu/ml/machine-learning-databases/'
piece2 = 'flags/flag.data'
url = piece1 + piece2
dataset_path = tf.keras.utils.get_file(ds, url)
dataset_path
```

- d. the following URL downloads information to identify column names:

<https://archive.ics.uci.edu/ml/machine-learning-databases/flags/flag.names>

(Download to get the column names and place them in a variable)

e. use the following code to read the dataset into pandas:

```
raw_dataset = pd.read_csv(dataset_path, names=cols)
```

(hint: consult Chapter 1 code example for basic assistance)

```
# we need the pandas library
import pandas as pd

cols = ['name', 'landmass', 'zone', 'area', 'population', 'language', 'religion',
        'bars', 'stripes', 'colours', 'red', 'green', 'blue', 'gold', 'white',
        'black', 'orange', 'mainhue', 'circles', 'crosses', 'saltires',
        'quarters', 'sunstars', 'crescent', 'triangle', 'icon', 'animate',
        'text', 'topleft', 'botright']

raw_dataset = pd.read_csv(dataset_path, names=cols)
```

f. display a few records from the dataframe to verify its contents.

```
raw_dataset.head()
```

g. convert dataframe into a TensorFlow dataset.

- i) remove all non-numerical columns from dataframe
(hint: use **pop** function)
- ii) use **from_tensor_slices** method
(hint: use `raw_dataset.values` as single parameter to method)
- iii) use **take** method to display one or more elements

```
raw_dataset.pop('name')
raw_dataset.pop('mainhue')
raw_dataset.pop('topleft')
raw_dataset.pop('botright')
```

```
dataset = tf.data.Dataset.from_tensor_slices(raw_dataset.values)
```

```
dataset = tf.data.Dataset.range(10)
```

4. Transform a tensor with the map method.

a. create a tensor:

```
dataset = tf.data.Dataset.range(10)
```

b. display the tensor:

```
for row in dataset:
    print (row)

tf.Tensor(0, shape=(), dtype=int64)
tf.Tensor(1, shape=(), dtype=int64)
tf.Tensor(2, shape=(), dtype=int64)
tf.Tensor(3, shape=(), dtype=int64)
tf.Tensor(4, shape=(), dtype=int64)
tf.Tensor(5, shape=(), dtype=int64)
tf.Tensor(6, shape=(), dtype=int64)
tf.Tensor(7, shape=(), dtype=int64)
tf.Tensor(8, shape=(), dtype=int64)
tf.Tensor(9, shape=(), dtype=int64)
```

c. repeat the tensor 5 times and batch place in batches of 4:

```
data_batch = dataset.repeat(5).batch(4)
```

d. display the batched tensor:

```
for row in data_batch:
    print (row)
```

e. use map to cube each element in the batched tensor:

```
data_map = data_batch.map(lambda x: x ** 3)
```

f. take two elements from the batched tensor and display:

```
for item in data_map.take(2):
    print (item)

tf.Tensor([ 0  1  8 27], shape=(4,), dtype=int64)
tf.Tensor([ 64 125 216 343], shape=(4,), dtype=int64)
```

5. Create a new dataset, batch, shuffle, and display.

a. new dataset:

```
dataset = tf.data.Dataset.range(10).repeat(5)
```

```
print ('dataset has', len(list(dataset)), 'elements')
```

b. batch into 7 elements with buffer size of 5, shuffle, and display:

```
ds = dataset.shuffle(buffer_size=5).batch(7)
```

```
for item in ds:  
    print (item)
```

```
tf.Tensor([4 2 1 5 6 0 3], shape=(7,), dtype=int64)  
tf.Tensor([8 7 3 0 9 6 2], shape=(7,), dtype=int64)  
tf.Tensor([1 9 8 1 5 0 3], shape=(7,), dtype=int64)  
tf.Tensor([2 5 7 7 4 0 9], shape=(7,), dtype=int64)  
tf.Tensor([2 6 3 1 8 4 5], shape=(7,), dtype=int64)  
tf.Tensor([7 0 6 9 2 1 8], shape=(7,), dtype=int64)  
tf.Tensor([4 3 4 6 8 7 9], shape=(7,), dtype=int64)  
tf.Tensor([5], shape=(1,), dtype=int64)
```