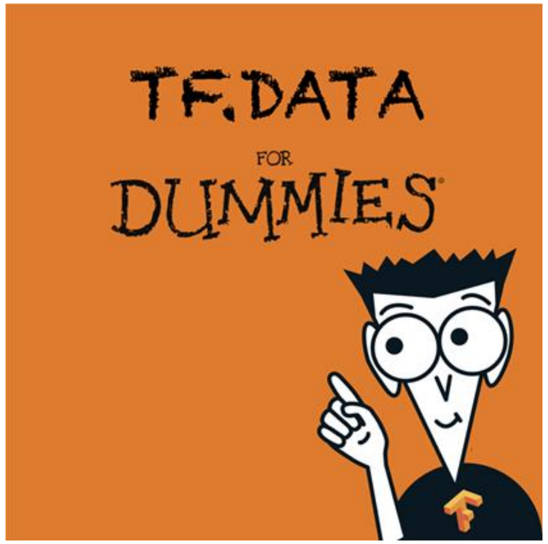
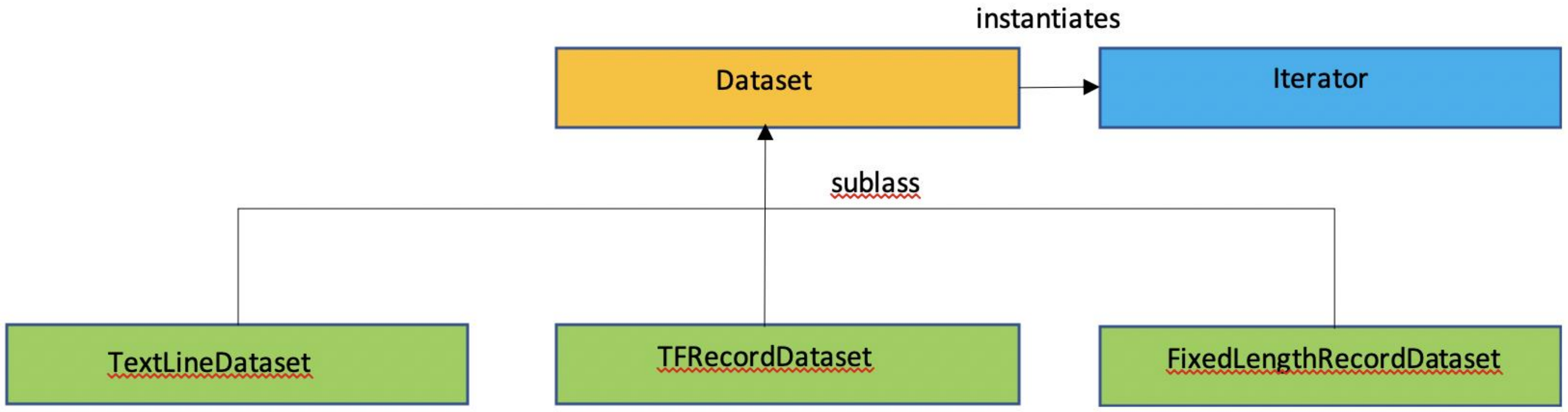
# 데이터 적재와 전처리

박 경 규

## 데이터 API

TensorFlow Data API takes a chunk of data, processes it, and passes it onto the next step of the workflow. The processing is generally called transformation.

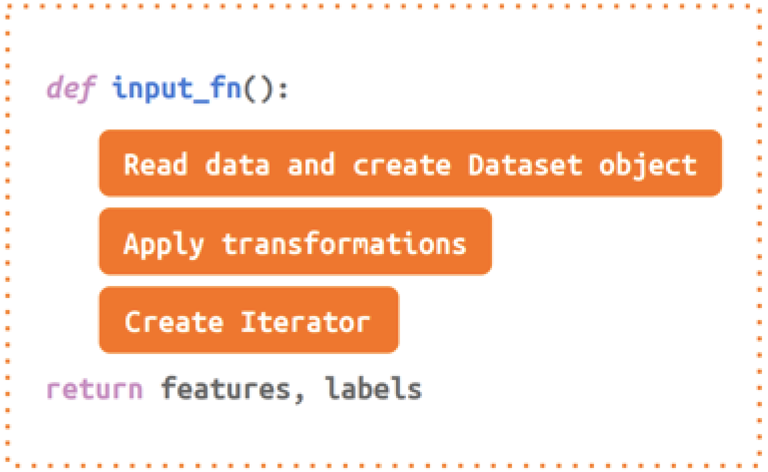




**1. Data Extraction**: The process of bringing data from an external location/previous step of workflow into the memory.
**2. Data Transformation:** This is the business logic if you may wherein the data brought in is processed.
**3. Data Load**: This step ensures that the transformed data is sent to the next step of the workflow/output location of your transformed data.

출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

## Data Extraction



```
def input_fn():
    Read data and create Dataset object
    Apply transformations
    Create Iterator
return features, labels
```

**1. Consuming NumPy arrays**

dataset = tf.data.Dataset.from_tensor_slices((images, labels))

**2. Consuming CSV files**

dataset = tf.data.experimental.make_csv_dataset( data_file, batch_size=4, label_name=”provide_label_name_for_shuffling”)

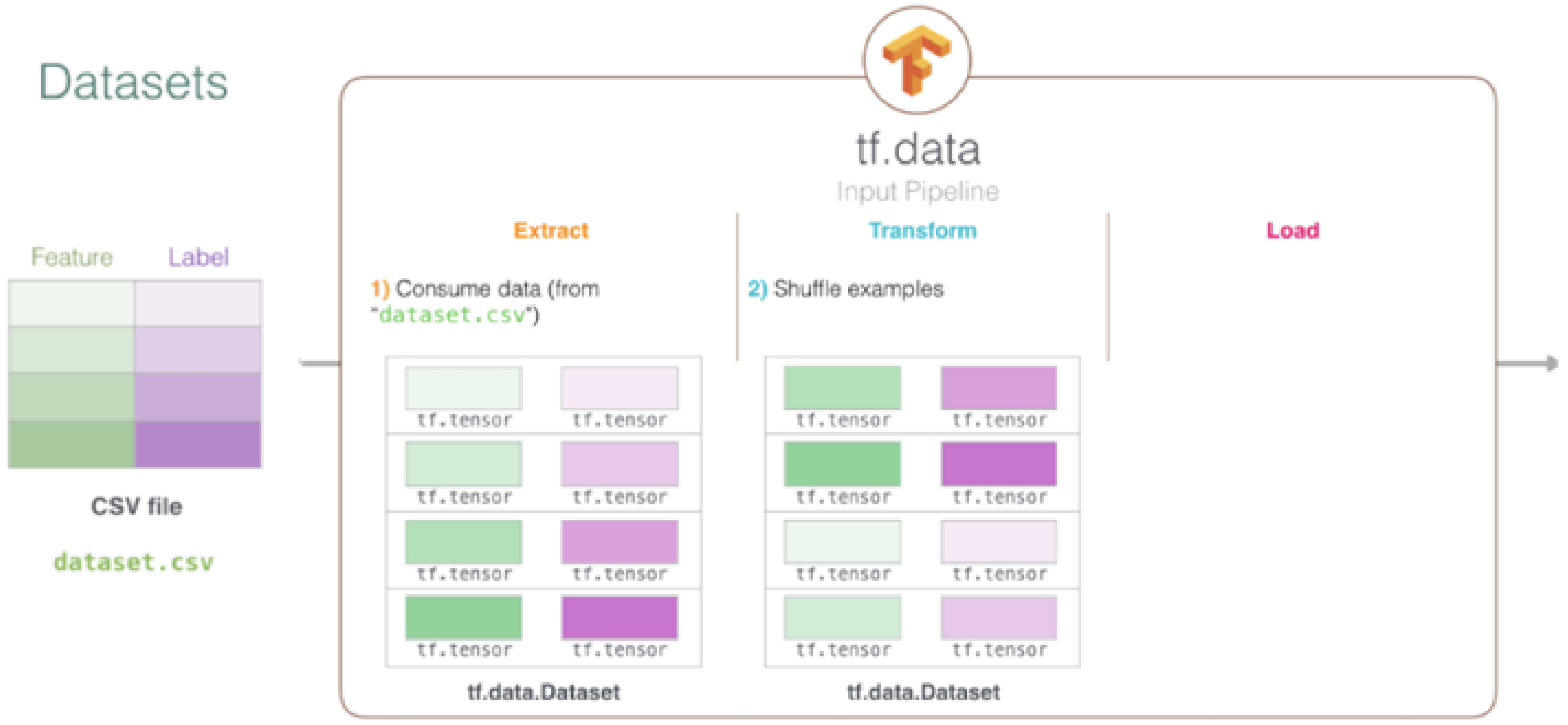
**3. Consuming Python Generators**

```
def get_numbers(limit):
    i = 0
while i<stop: yield i i += 1 ds_counter = tf.data.Dataset.from_generator(count, args=[25], output_types=tf.int32, output_shapes = (), )
```

출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

## Data Transformation - Shuffling

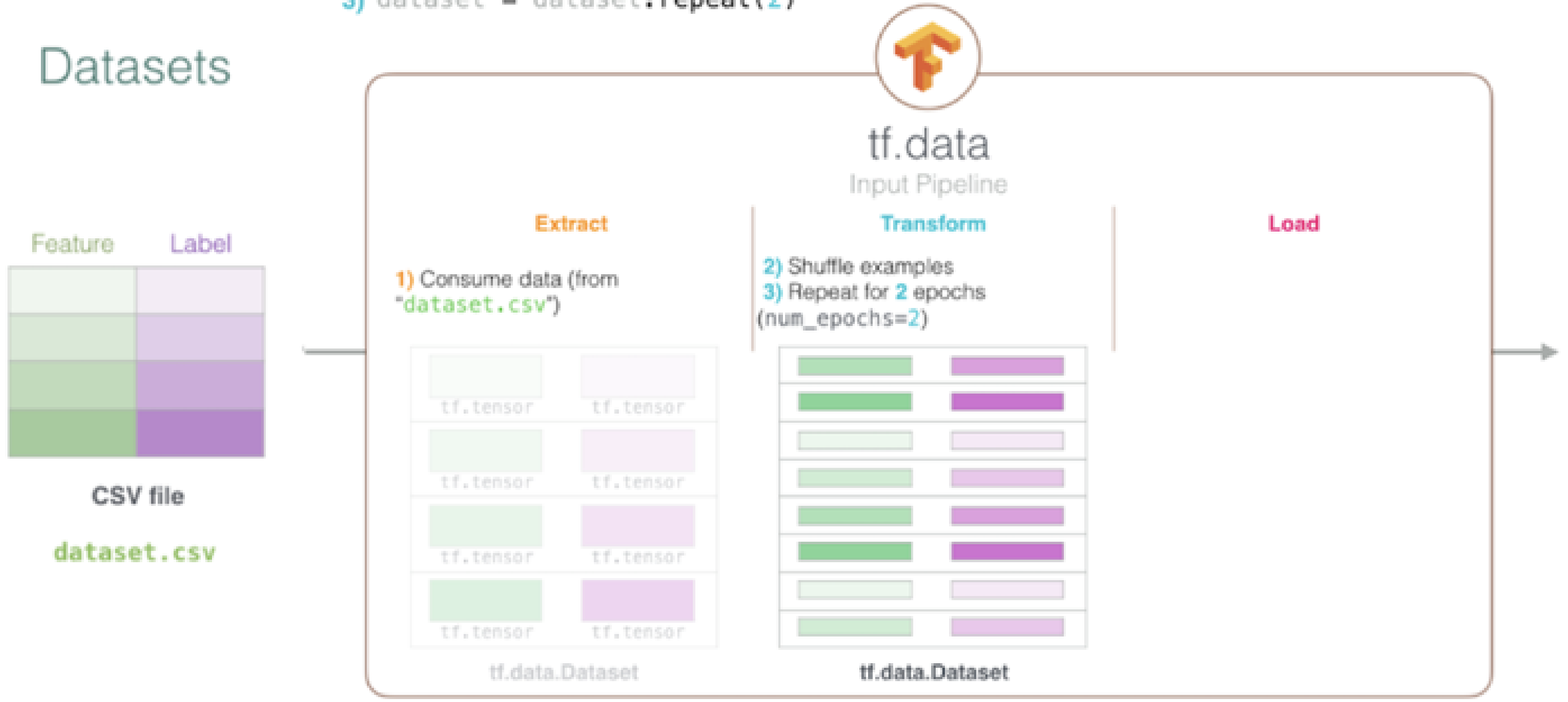
```
1) dataset = tf.data.experimental.CsvDataset("dataset.csv", record_defaults)
2) dataset = dataset.shuffle(10000)
```



출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

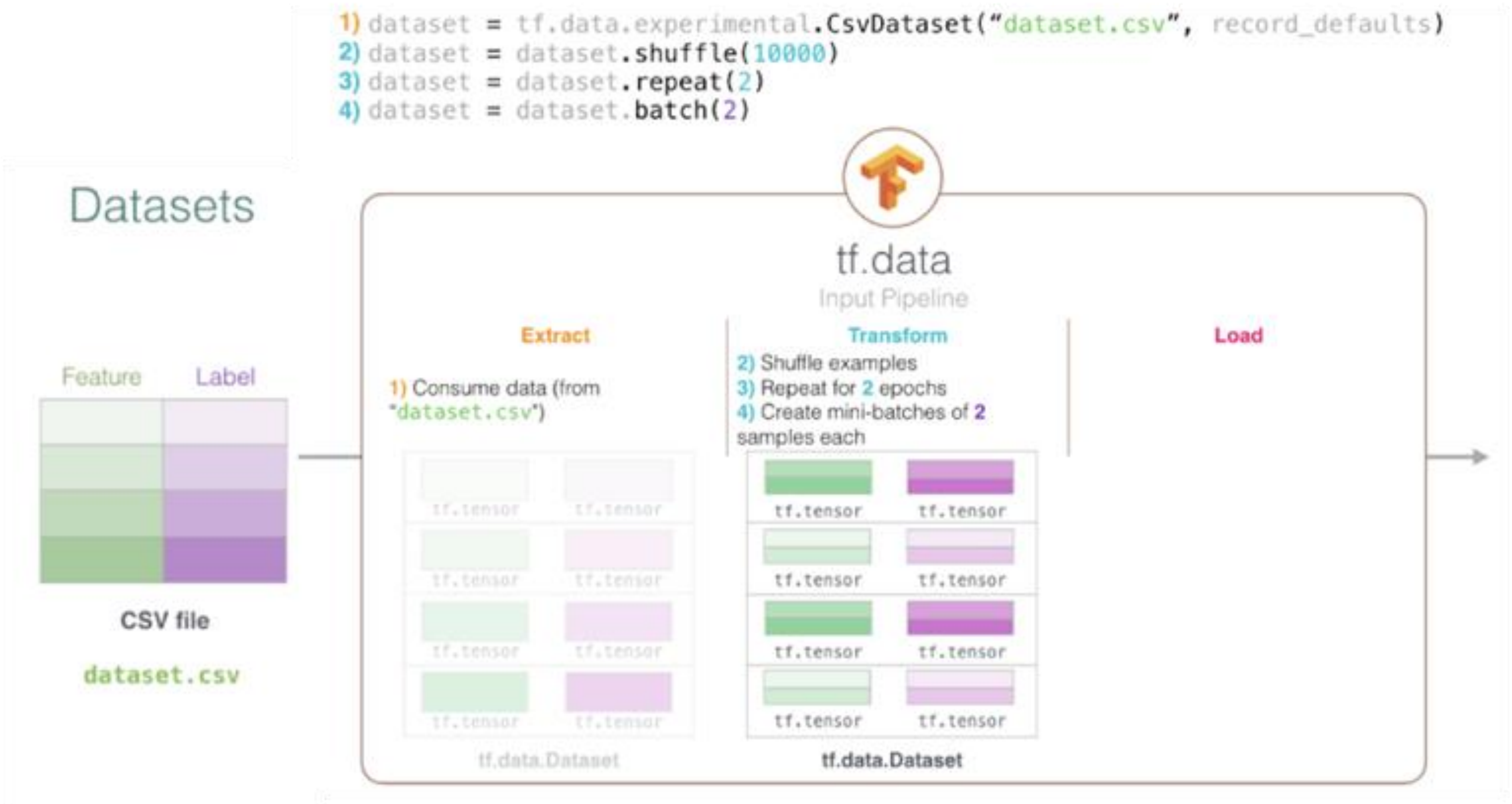
## Data Transformation - Repeat

```
1) dataset = tf.data.experimental.CsvDataset("dataset.csv", record_defaults)
2) dataset = dataset.shuffle(10000)
3) dataset = dataset.repeat(2)
```



출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

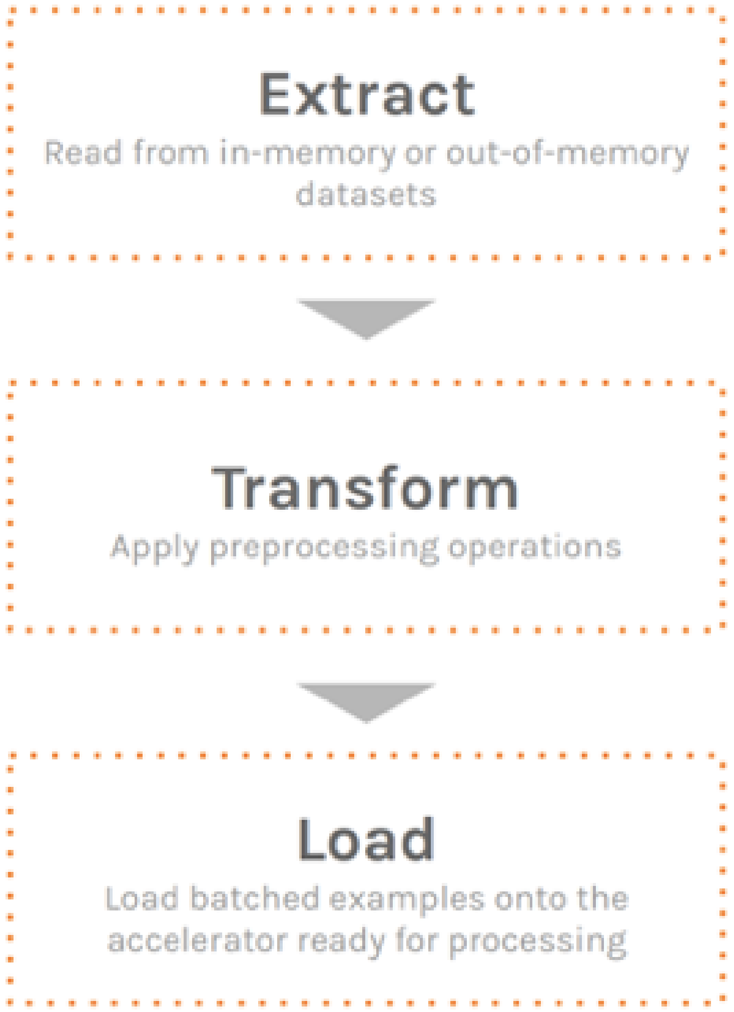
## Data Transformation - Batching



```
1) dataset = tf.data.experimental.CsvDataset("dataset.csv", record_defaults)
2) dataset = dataset.shuffle(10000)
3) dataset = dataset.repeat(2)
4) dataset = dataset.batch(2)
```

출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

## Data Load



Extract
Read from in-memory or out-of-memory datasets

Transform
Apply preprocessing operations

Load
Load batched examples onto the accelerator ready for processing

Methods to create a Dataset object from a data source:
+ tf.data.Dataset.from_tensor_slices
+ tf.data.Dataset.from_generator
+ tf.data.TFRecordDataset
+ tf.data.TextLineDataset

Methods to transform a Dataset:
+ tf.data.Dataset.batch
+ tf.data.Dataset.shuffle
+ tf.data.Dataset.map
+ tf.data.Dataset.repeat

Prefetch elements from the input Dataset ahead of the time they are requested by calling the tf.data.Dataset.prefetch method. This transformation overlaps the work of a producer and consumer.

출처 : https://medium.com/swlh/tensorflow-data-for-dummies-1132cd002e8f

## 데이터 API

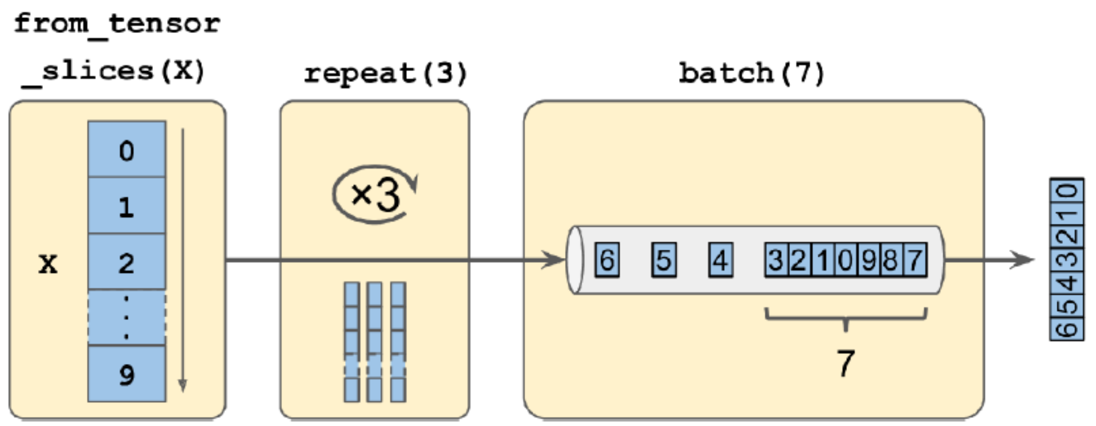
아주 큰 데이터셋으로 딥러닝 시스템을 훈련해야 하는 경우가 많으며 텐서플로 데이터 API로 쉽게 처리할 수 있습니다.

### ■ 데이터셋(dataset)

```
>>> X = tf.range(10)  # any data tensor
>>> dataset = tf.data.Dataset.from_tensor_slices(X)
>>> dataset
<TensorSliceDataset shapes: (), types: tf.int32>
```

```
>>> for item in dataset:
...     print(item)
...
tf.Tensor(0, shape=(), dtype=int32)
tf.Tensor(1, shape=(), dtype=int32)
tf.Tensor(2, shape=(), dtype=int32)
[...]
tf.Tensor(9, shape=(), dtype=int32)
```

```
>>> dataset = dataset.repeat(3).batch(7)
>>> for item in dataset:
...     print(item)
...
tf.Tensor([0 1 2 3 4 5 6], shape=(7,), dtype=int32)
tf.Tensor([7 8 9 0 1 2 3], shape=(7,), dtype=int32)
tf.Tensor([4 5 6 7 8 9 0], shape=(7,), dtype=int32)
tf.Tensor([1 2 3 4 5 6 7], shape=(7,), dtype=int32)
tf.Tensor([8 9], shape=(2,), dtype=int32)
```



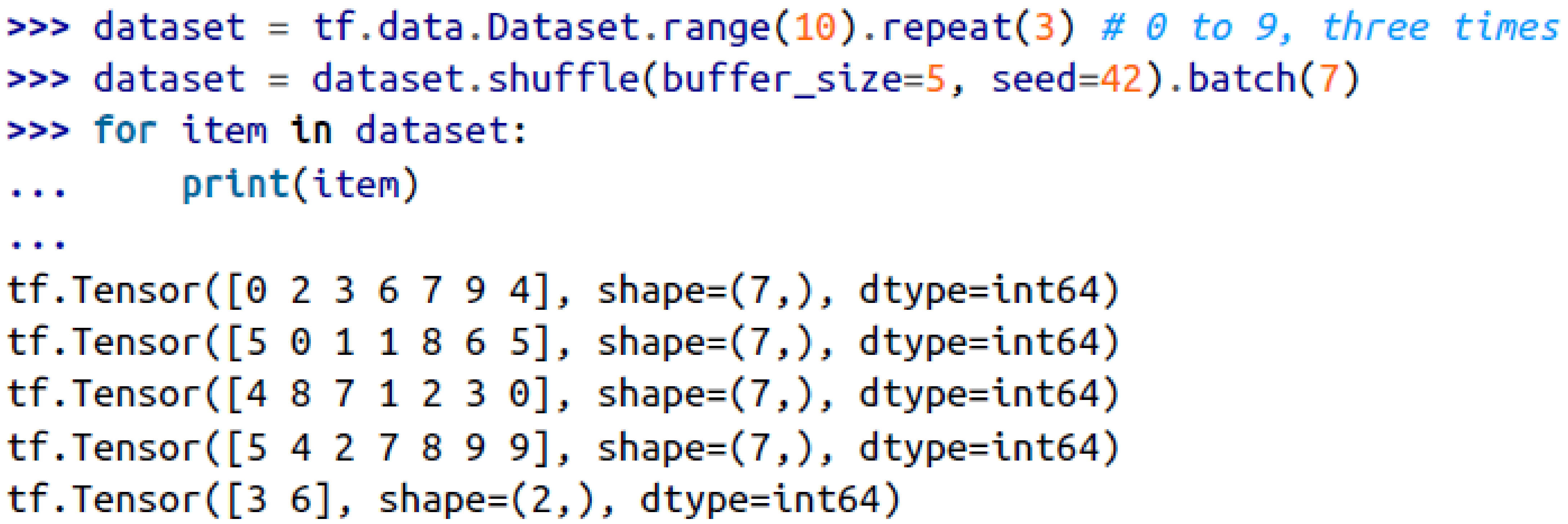
출처 : 핸즈온 머신러닝 2판(Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition)

## 연쇄 변환

```
import numpy as np
import tensorflow as tf
from tensorflow import keras

# Datasets
X = tf.range(10)
dataset = tf.data.Dataset.from_tensor_slices(X)
print(dataset)

for item in dataset:
    print(item)

# 연쇄 변환
dataset = dataset.repeat(3).batch(7)
for item in dataset:
    print(item)

dataset = dataset.map(lambda x: x * 2)
for item in dataset:
    print(item)

dataset = dataset.unbatch()
for item in dataset:
    print(item)

dataset = dataset.filter(lambda x: x < 10)  # keep only items < 10
for item in dataset:
    print(item)

for item in dataset.take(3):
    print(item)
```

## 데이터 셔플링

경사하강법은 훈련 세트에 있는 샘플이 독립적이고 동일한 분포일 때 최고의 성능을 발휘하므로 shuffle() 메서드로 샘플을 섞는 것입니다. 버퍼 크기를 충분히 크게 하는 것이 중요합니다.


```
>>> dataset = tf.data.Dataset.range(10).repeat(3) # 0 to 9, three times
>>> dataset = dataset.shuffle(buffer_size=5, seed=42).batch(7)
>>> for item in dataset:
...     print(item)
...
tf.Tensor([0 2 3 6 7 9 4], shape=(7,), dtype=int64)
tf.Tensor([5 0 1 1 8 6 5], shape=(7,), dtype=int64)
tf.Tensor([4 8 7 1 2 3 0], shape=(7,), dtype=int64)
tf.Tensor([5 4 2 7 8 9 9], shape=(7,), dtype=int64)
tf.Tensor([3 6], shape=(2,), dtype=int64)
```

# Thank you

박경규, 010-7240-1128
kgpark88@gmail.com