coursera Help

Parallelization with TFDS

In this week's exercise, we'll go back to the classic cats versus dogs example, but instead of just naively loading the data to train a model, you will be parallelizing various stages of the Extract, Transform and Load processes. In particular, you will be performing following tasks:

1. Parallelize the extraction of the stored TFRecords of the cats_vs_dogs dataset by using the interleave operation.
2. Parallelize the transformation during the preprocessing of the raw dataset by using the map operation.
3. Cache the processed dataset in memory by using the cache operation for faster retrieval.
4. Parallelize the loading of the cached dataset during the training cycle by using the prefetch operation.

Setup

In [1]:
```
import multiprocessing

import tensorflow as tf
import tensorflow_datasets as tfds

from os import getcwd
```

Create and Compile the Model

In [3]:
```
def create_model():
    input_layer = tf.keras.layers.Input(shape=(224, 224, 3))
    base_model = tf.keras.applications.MobileNetV2(input_tensor=input_layer,
                                                   weights='imagenet',
                                                   include_top=False)
    base_model.trainable = False
    x = tf.keras.layers.GlobalAveragePooling2D()(base_model.output)
    x = tf.keras.layers.Dense(2, activation='softmax')(x)

    model = tf.keras.models.Model(inputs=input_layer, outputs=x)
    model.compile(optimizer='adam',
                  loss='sparse_categorical_crossentropy',
                  metrics=['acc'])
    return model
```

Naive Approach

Just for comparison, let's start by using the naive approach to Extract, Transform, and Load the data to train the model defined above. By naive approach we mean that we won't apply any of the new concepts of parallelization that we learned about in this module.