Congratulations! You passed!

Grade received 88.89% **To pass** 80% or higher

Go to next item

1. What does the acronym ETL stand for? **1 / 1 point**

- ○ Extract / Transfer / Load
- ○ External / Transform / Load
- ○ Enhance / Transfer / Load
- ◉ Extract / Transform / Load

✓ **Correct**

2. You have a multi processor machine, containing a CPU and GPU. How would you typically distribute these for training a model? **0 / 1 point**

- ○ Use CPU for extracting and loading, and the GPU for transforming
- ○ Use CPU for extracting, transferring and loading, and the GPU for training
- ○ Use CPU and GPU for all tasks in parallel
- ◉ Use CPU for extracting and the GPU for transforming and loading

✗ **Incorrect**

3. One way to speed up ETL is to use a cache. What's the API for this called? **1 / 1 point**

- ○ tf.data.DataCache()
- ○ tf.data.Dataset.ETLCache()
- ○ tf.data.Dataset.datacache()
- ◉ tf.data.Dataset.cache()

✓ **Correct**

4. I have a dataset loaded using this code: **1 / 1 point**

```
dataset = tfds.load('cats_vs_dogs',split=tfds.Split.TRAIN)
```

How would I cache it on disk?

- ○ train_dataset = dataset.cache(file='cache')
- ○ train_dataset = dataset.cache()
- ○ train_dataset = dataset.cache(cachename=file)
- ◉ train_dataset = dataset.cache(filename='cache')

✓ **Correct**

5. I have a dataset loaded using this code: **1 / 1 point**

```
dataset = tfds.load('cats_vs_dogs',split=tfds.Split.TRAIN)
```

How would I cache it in memory?

- ○ train_dataset = dataset.cache(cachename='memory')
- ○ train_dataset = dataset.memorycache()
- ○ train_dataset = dataset.cache_in_memory()
- ◉ train_dataset = dataset.cache()

✓ **Correct**

6. If I create a function called 'augment' that transforms data, what code would I use to apply this after loading a dataset with **1 / 1 point**

```
dataset = tfds.load('cats_vs_dogs',split=tfds.Split.TRAIN)
```

- ○ augmented_dataset = dataset.augment()
- ○ augmented_dataset = map(augment)
- ○ augmented_dataset = dataset.augment(dataset)
- ◉ augmented_dataset = dataset.map(augment)

Correct

7. If you want to parallelise the transform of a dataset across multiple cores, what's the correct call? **1 / 1 point**

- s = dataset.map(augment, parallel_calls=2)
- s = dataset.map(augment, num_parallel=2)
- s = dataset.map(augment, 2)
- [x] s = dataset.map(augment, num_parallel_calls=2)

Correct

8. If you're not sure how many cores are accessible, for example, if you're running in a shared cloud environment, how can you find out how many are available to you? **1 / 1 point**

- num_cores = multiprocessing.available_cpus()
- [x] num_cores = multiprocessing.cpu_count()
- It's not possible
- num_cores = multiprocessing.cpu.count()

Correct

9. The process of executing a custom map function over a batch of inputs is called: **1 / 1 point**

- Batch mapping
- [x] Vectorization
- Map batching
- Visualization

Correct