TFRecords

Learn how protocol buffers are stored in TFRecords files.

Chapter Goals:

- Learn how to write serialized protocol buffers to TFRecords files
- Implement a function that writes a list of feature data to a TFRecords file

A. Serialization

After creating a `tf.train.Example` protocol buffer, we normally store it in a file. To do this, we first have to *serialize* the object, i.e. convert it to a byte string which can be written to a file. The way we serialize a `tf.train.Example` object is through its `SerializeToString` method.

```
import tensorflow as tf

ex = tf.train.Example(features=tf.train.Features(feature=f_dict))

print(repr(ex))

ser_ex = ex.SerializeToString()
print(ser_ex)
```

B. Writing to data files

We store serialized `tf.train.Example` protocol buffers in special files called TFRecords files. The simple way to write to a TFRecords file is through a `TFRecordWriter`.

```
import tensorflow as tf

writer = tf.python_io.TFRecordWriter('out.tfrecords')
writer.write(ser_ex)
writer.close()
```

The `TFRecordWriter` is initialized with the output file that it writes to. In our example, we wrote to `'out.tfrecords'`.

The `write` function takes in a byte string and writes that byte string to the end of the output file. After we're done writing to the output file, we close the file using the `close` function.

You can also write multiple serialized `tf.train.Example` objects to a single file, as long as the file is open.

```
import tensorflow as tf

# Writing 3 Example objects to the same file
writer = tf.python_io.TFRecordWriter('out.tfrecords')
writer.write(ser_ex1)
writer.write(ser_ex2)
writer.write(ser_ex3)
writer.close()
```