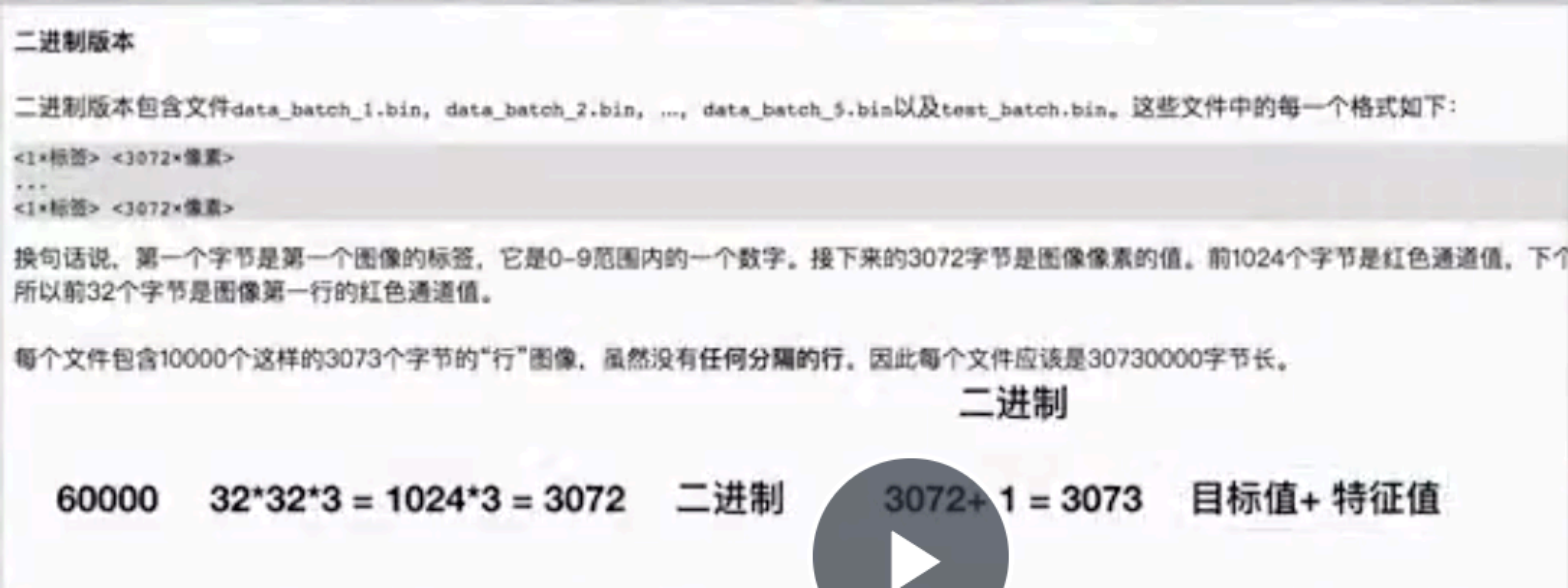
3、文件读取API-文件内容解码器

- 由于从文件中读取的是字符串，需要函数去解析这些字符串到张量

- tf.decode_csv(records,record_defaults=None,field_delim = None，name = None)
  将CSV转换为张量，与tf.TextLineReader搭配使用
  - records:tensor型字符串，每个字符串是csv中的记录行
  - field_delim:默认分割符","
  - record_defaults:参数决定了所得张量的类型，并设置一个值在输入字符串中缺少使用默认值,如
- tf.decode_raw(bytes,out_type,little_endian = None，name = None)
  将字节转换为一个数字向量表示，字节为一字符串类型的张量,与函数tf.FixedLengthRecordReader搭配使用，二进制读取为uint8格式



二进制版本

二进制版本包含文件data_batch_1.bin, data_batch_2.bin, …, data_batch_5.bin以及test_batch.bin。这些文件中的每一个格式如下：

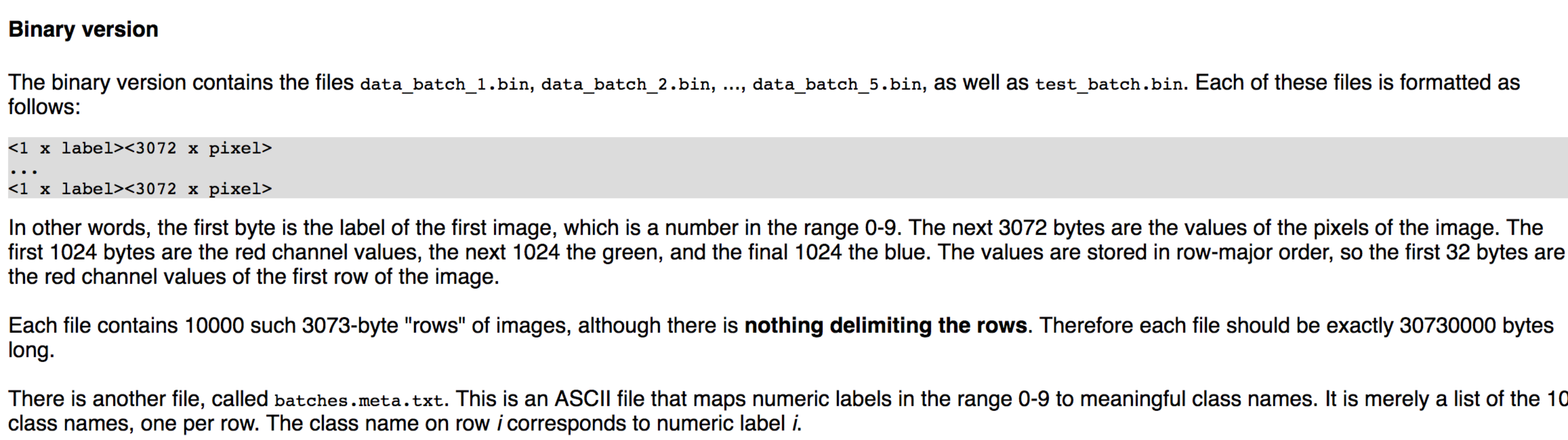
```
<1×标签> <3072×像素>
...
<1×标签> <3072×像素>
```

换句话说，第一个字节是第一个图像的标签，它是0-9范围内的一个数字。接下来的3072字节是图像像素的值。前1024个字节是红色通道值，下个
所以前32个字节是图像第一行的红色通道值。

每个文件包含10000个这样的3073个字节的“行”图像，虽然没有任何分隔的行。因此每个文件应该是30730000字节长。

二进制

60000   32*32*3 = 1024*3 = 3072   二进制   3072+ 1 = 3073   目标值+ 特征值



Binary version

The binary version contains the files `data_batch_1.bin`, `data_batch_2.bin`, ..., `data_batch_5.bin`, as well as `test_batch.bin`. Each of these files is formatted as follows:

```
<1 x label><3072 x pixel>
...
<1 x label><3072 x pixel>
```

In other words, the first byte is the label of the first image, which is a number in the range 0-9. The next 3072 bytes are the values of the pixels of the image. The first 1024 bytes are the red channel values, the next 1024 the green, and the final 1024 the blue. The values are stored in row-major order, so the first 32 bytes are the red channel values of the first row of the image.

Each file contains 10000 such 3073-byte "rows" of images, although there is **nothing delimiting the rows**. Therefore each file should be exactly 30730000 bytes long.

There is another file, called `batches.meta.txt`. This is an ASCII file that maps numeric labels in the range 0-9 to meaningful class names. It is merely a list of the 10 class names, one per row. The class name on row *i* corresponds to numeric label *i*.