文件传给大数据平台，文件格式是GBK编码， 使用spark的外部数据源、textFile等方式读取文件都是UTF-8格式， 导致中文乱码。

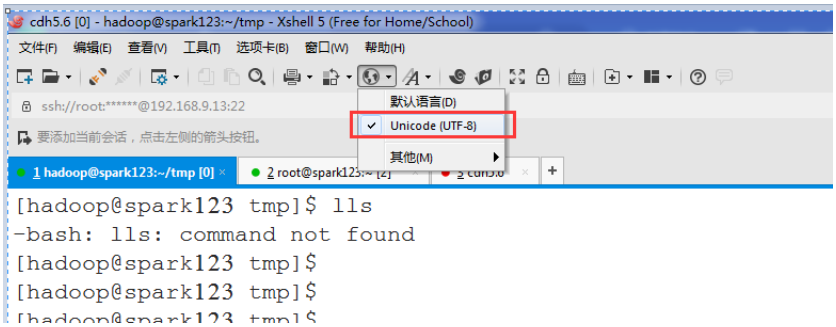
1. 解决文本格式的文件的乱码

使用spark的textFile读取文本文件， 但是这个方法只支持UTF8格式， 如果GBK格式的文件含有中文， 就会乱码。

生成测试数据：

（1）、生成utf8格式的测试数据， 并上传到hdfs

先将ssh工具的编码修改为utf8：



生成数据：

```
$ echo "安徽|合肥|0551" >> city_utf8.txt
$ echo "江苏|南京|025" >> city_utf8.txt
$ echo "浙江|杭州|0571" >> city_utf8.txt
$ cat city_utf8.txt
安徽|合肥|0551
江苏|南京|025
浙江|杭州|0571
$
$ hdfs dfs -put city_utf8.txt /tmp/input/city_utf8.txt
```

（2）、将utf8格式的数据转换成gbk格式， 并上传到hdfs

```
$ iconv -f UTF-8 -t GBK city_utf8.txt > city_gbk.txt
$ hdfs dfs -put city_gbk.txt /tmp/input/city_gbk.txt
```

Scala代码：

```
```

```scala
package test

import org.apache.hadoop.io.{LongWritable, Text}
import org.apache.hadoop.mapred.TextInputFormat
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.sql.hive.HiveContext

/**
  * Created on 下午5:58.
  * desc: spark 解析中文文件乱码
  * @author hadoop
  */
object testSparkEncoding {
  def main(args: Array[String]): Unit = {

    val sparkConf = new SparkConf().setMaster("local[3]").setAppName("test")
    val sc = new SparkContext(sparkConf)
    val sqlContext = new HiveContext(sc)


    ////////////////////////////////////////////////////////////////////////////////
    //
    // 使用textfile读取utf8格式的文件
    //  输出:
    //    安徽|合肥|0551
    //    江苏|南京|025
    //    浙江|杭州|0571

    ////////////////////////////////////////////////////////////////////////////////
    val fileUTF8 = "/tmp/input/city_utf8.txt"
    val rddUTF8 = sc.textFile(fileUTF8)  // 读取文件
    rddUTF8.take(10).foreach(println)   // 打印文件的前10行内容


    ////////////////////////////////////////////////////////////////////////////////
    //
    // 使用textfile读取GBK格式的文件
    //  输出乱码:
    //    ????|?Þ?|0551
    //    ????|?G?|025
    //    ?暿|????|0571

    ////////////////////////////////////////////////////////////////////////////////
```

```
    /////
    val fileGBK1 = "/tmp/input/city_gbk.txt"
    val rddGBK1 = sc.textFile(fileGBK1)    // 读取文件
    rddGBK1.take(10).foreach(println)    // 打印文件的前10行内容


    ////////////////////////////////////////////////////////////////////////////////
    /////
    //
    // 读取GBK格式乱码处理
    //  输出:
    //    安徽|合肥|0551
    //    江苏|南京|025
    //    浙江|杭州|0571

    ////////////////////////////////////////////////////////////////////////////////
    /////
    val fileGBK2 = "/tmp/input/city_gbk.txt"
    val rddGBK2 = sc.hadoopFile(fileGBK2, classOf[TextInputFormat],
classOf[LongWritable], classOf[Text], 1).
      map(p => new String(p._2.getBytes, 0, p._2.getLength, "GBK"))
    rddGBK2.take(10).foreach(println)
  }
}
```

2. 解决json格式文件的乱码

spark的外部数据源可以解析json文件的格式， 如果自己解析json格式数据，自己解析json文件难度比较大（需要考虑每行的schema是否一致、解析失败等）。 但是spark解析json的方法也只支持utf8格式。

测试数据：
同样的方式生成GBK的数据：
{"pro":"安徽","city":"合肥", "code":"0551"}
{"pro":"江苏","city":"南京", "code":"025"}
{"pro":"浙江","city":"杭州", "code":"0571"}

（1）、使用外部数据源读取

```
package test

import org.apache.hadoop.io.{LongWritable, Text}
import org.apache.hadoop.mapred.TextInputFormat
import org.apache.spark.sql.hive.HiveContext
import org.apache.spark.{SparkConf, SparkContext}

/**
  * Created on 下午5:58.
  * desc: spark 解析中文文件乱码
  *
  * @author hadoop
  */
object testSparkJsonEncoding {
  def main(args: Array[String]): Unit = {
    val sparkConf = new SparkConf().setMaster("local[3]").setAppName("test")
    val sc = new SparkContext(sparkConf)
    val sqlContext = new HiveContext(sc)


    ////////////////////////////////////////////////////////////////////////////////
    //
    // 使用spark的外部数据源读取json格式的文件

    ////////////////////////////////////////////////////////////////////////////////
    val jsonFile = "/tmp/input/test.json"
    val jsonDF = sqlContext.read.format("json").load(jsonFile)
    jsonDF.show()
  }
}

```

中文字段乱码：

```
+----+----+----+
|city|code| pro|
+----+----+----+
| ??|0551|????|
| ?c?| 025|????|
|????|0571|  ??|
+----+----+----+
```

（2）、处理乱码

jsonRDD是通过hadoopFile将编码转成UTF8， 返回的数据类型是：RDD[String]，将这个变量传入：sqlContext.read.json(jsonRDD)

```
1  package test
2
3  import org.apache.hadoop.io.{LongWritable, Text}
4  import org.apache.hadoop.mapred.TextInputFormat
5  import org.apache.spark.sql.hive.HiveContext
6  import org.apache.spark.{SparkConf, SparkContext}
7
8  /**
9    * Created on 下午5:58.
10   * desc: spark 解析中文文件乱码
11   *
12   * @author zhoucw
13   */
14 object testSparkJsonEncoding {
15   def main(args: Array[String]): Unit = {
16     val sparkConf = new SparkConf().setMaster("local[3]").setAppName("test")
17     val sc = new SparkContext(sparkConf)
18     val sqlContext = new HiveContext(sc)
19
20
    ////////////////////////////////////////////////////////////////////////////////
   /////
21     //
22     // 使用spark的外部数据源读取json格式的文件，处理中文乱码
23     //
24
    ////////////////////////////////////////////////////////////////////////////////
   /////
25     val jsonRDD =
   sc.hadoopFile(jsonFile,classOf[TextInputFormat],classOf[LongWritable],classOf[
   Text],1).map(p => new String(p._2.getBytes, 0, p._2.getLength, "GBK"))
26     val jsonDF2 = sqlContext.read.json(jsonRDD)
```

```
27      jsonDF2.show()
28    }
29  }
30
```

中文可以正常显示：

```
+----+----+---+
|city|code|pro|
+----+----+---+
|  合肥|0551| 安徽|
|  南京| 025| 江苏|
|  杭州|0571| 浙江|
+----+----+---+
```