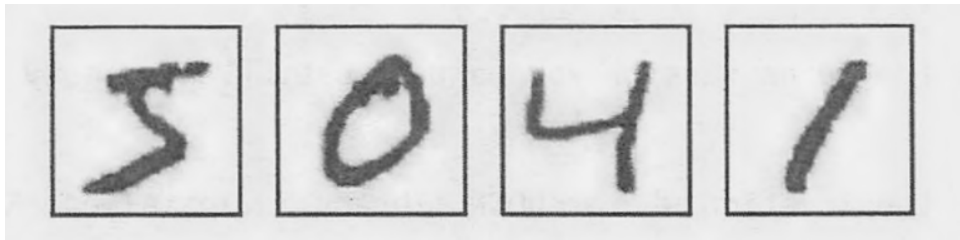


TensorFlow | 1 | 识别手写数字

1. 基本环境

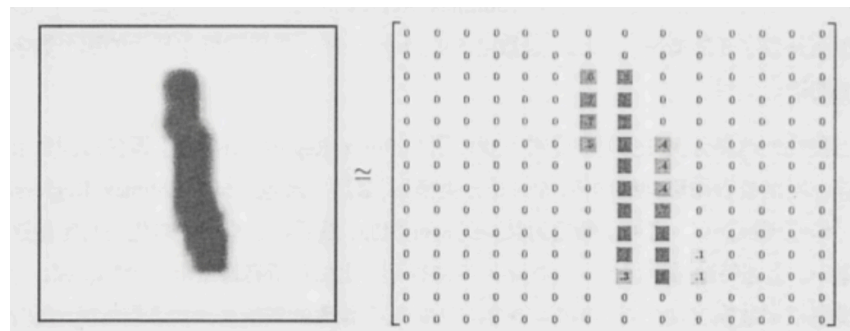
| | |
|------|---------------------------|
| 程序名称 | ex1_1024.py |
| 硬件环境 | MacOSX |
| 软件环境 | Python3, Spyder, Anaconda |

2. 核心问题：利用 TensorFlow 识别手写数字，基于《TensorFlow 实战》MNIST(Mixed National Institute of Standards and Technology database)是一个机器视觉数据集，如图所示，它由几万张 28 像素×28 像素的手写数字组成，这些图片只包含灰度值信息，我们的任务就是对这些图片进行分类识别，进而来探索 TensorFlow。

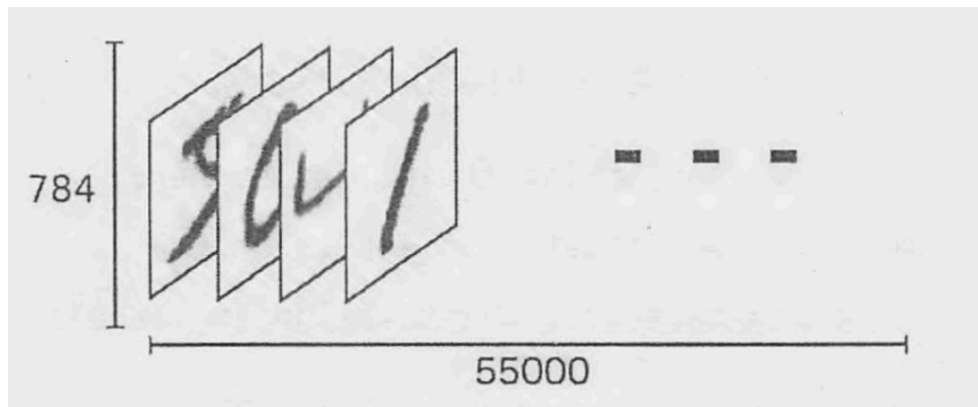


3. 解决过程：

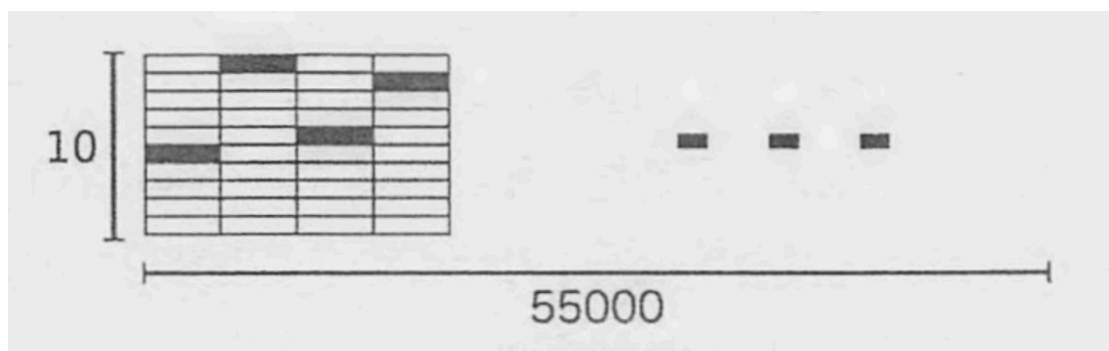
- ✖ 首先对 MNIST 数据进行加载。
- ✖ 然后查看 minst 这个数据集的情况，可以看到训练集有 55000 个样本，测试集有 10000 个样本，同时验证集上有 5000 个样本。每一个样本都有它对应的标注信息，即 label。我们将在训练集上训练模型，在验证集上检验效果并决定何时完成训练，最后我们在测试集评测模型的效果（可通过准确率、召回率、F1-score 等评测）。
- ✖ 如下图，灰度图像空白部分全部为 0，有笔迹的地方根据颜色的深浅在 0 到 1 之间取值。每个样本有 $28 \times 28 = 784$ 维的特征，也就是 784 个点展开成 1 维的结果，每一张图片用同样的顺序展开。



- ✖ 我们训练数据的特征是一个 55000×784 的 Tensor，第一个维度是图片的编号，第二个维度是图片中像素点的编号，如下图所示。



- ✖ 如下图，这里对 10 个种类进行了 one-hot 编码，Label 是一个 10 维的向量。只有 1 个值为 1，其余为 0。比如数字 0，对应的 Label 就是 $[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ ，数字 5 对应的 Label 就是 $[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]$ ，数字 n 就代表对应位置的值为 1。

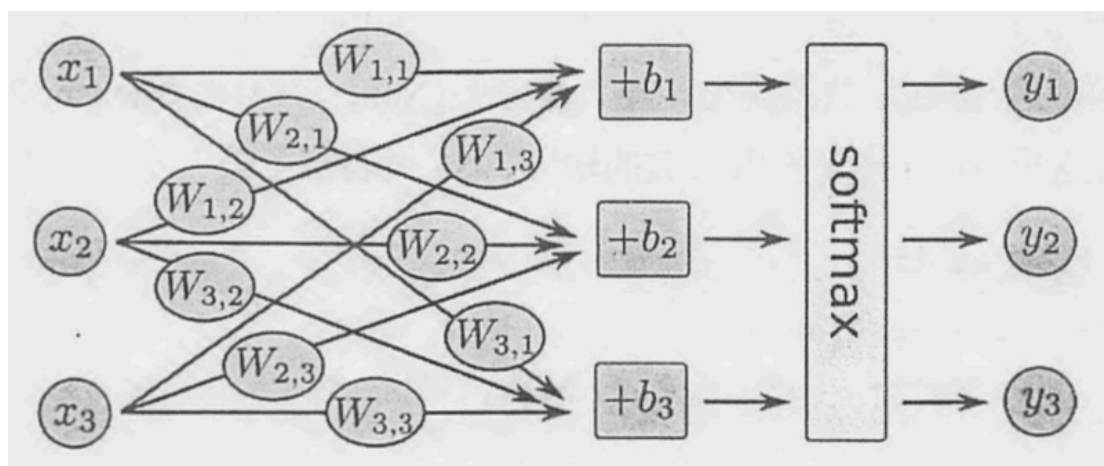


- ✖ 准备好数据后，接下来是设计算法。当我们处理多分类任务时，通常需要使用 Softmax Regression 模型。它的工作原理为，将可以判定为某类的特征相加，然后将这些特征转化为判定是这一类的概率。

$$\text{特征 feature}_i = \sum_j W_{i,j} X_j + b_i$$

- ✖ 接下来对所有的特征计算 $\text{Softmax}(\mathbf{x}) = \text{normalize}(\exp(\mathbf{x}))$ ，其中判定为第 i 类的概率就可由下面的公式得到 $\text{Softmax}(\mathbf{x})_t = \frac{\exp(x_t)}{\sum_j \exp(x_j)}$

把整个计算过程可视化：



上述可以表达为： $y = \text{softmax}(Wx + b)$

✱ 接下来就是使用 TensorFlow 实现一个 Softmax Regression。

(1) 首先载入 TensorFlow 库，并创建一个新的 InteractiveSession。

```
import tensorflow as tf
sess = tf.InteractiveSession()
x = tf.placeholder(tf.float32, [None, 784])
```

(2) 接下来要给 Softmax Regression 模型中的 weights 和 biases 创建 Variable 对象。

```
W = tf.Variable(tf.zeros([784, 10]))
b = tf.Variable(tf.zeros([10]))
```

(3) 接下来就是要实现 Softmax Regression 算法。

```
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

(4) 在 TensorFlow 中定义 cross-entropy。

```
y_ = tf.placeholder(tf.float32, [None, 10])
cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ *
    tf.log(y), reduction_indices=[1]))
```

(5) 下一步使用 TensorFlow 的全局参数初始化器

```
tf.global_variables_initializer, 并执行它的 run 方法。
tf.global_variables_initializer().run()
```

(6) 最后，开始迭代地执行训练操作 train_step。

```
for i in range(1000)
    batch_xs, batch_ys = mnist.train.next_batch(100)
```

```
train_step.run({x:batch_xs, y_: batch_ys})
```

4. 代码：

```
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/",
one_hot=True)
sess=tf.InteractiveSession()
x=tf.placeholder(tf.float32,[None,784])
W=tf.Variable(tf.zeros([784,10]))
b=tf.Variable(tf.zeros([10]))
y=tf.nn.softmax(tf.matmul(x,W)+b)
y_=tf.placeholder(tf.float32,[None,10])

cross_entropy=tf.reduce_mean(-tf.reduce_sum(y_
*tf.log(y),reduction_indices=[1]))
train_step=tf.train.GradientDescentOptimizer(0.5).minimize
(cross_entropy)
tf.global_variables_initializer().run()
for i in range(1000):
    batch_xs,batch_ys=mnist.train.next_batch(100)
    train_step.run({x:batch_xs,y_:batch_ys})
correct_pre=tf.equal(tf.argmax(y,1),tf.argmax(y_,1))
accu=tf.reduce_mean(tf.cast(correct_pre,tfloat32))
print(accu.eval({x:mnist.test.images,y_:mnist.test.labels
}))
```

5. 结果：

准确率为 0.91 左右，由于只使用了最基础的模型，故准确率不高。

通过这个例子，我们使用 TensorFlow 实现了一个简单的机器学习算法 Softmax Regression，整个流程可以分为四部分：

- (1) 定义算法公式，也就是神经网络 forward 时的计算。
- (2) 定义 loss，选定优化器，并制定优化器优化 loss。

- (3) 迭代地对数据进行训练。
- (4) 在测试集或验证集上对准确率进行评测。

6. 报错记录

输入 `"mnist = input_data.read_data_sets('MNIST_data/', one_hot=True)"`

报错原因：网页找不到

解决方法：翻墙

log:

Extracting MNIST_data/train-images-idx3-ubyte.gz

Traceback (most recent call last):

```
File "<ipython-input-4-6f78a56b35f2>", line 1, in <module>
    mnist = input_data.read_data_sets('MNIST_data/',
one_hot=True)
```

```
File      "/Users/xuyizhou/anaconda3/lib/python3.6/site-
packages/tensorflow/contrib/learn/python/learn/datasets/m
nist.py", line 240, in read_data_sets
```

```
    SOURCE_URL + TRAIN_LABELS)
```

```
File      "/Users/xuyizhou/anaconda3/lib/python3.6/site-
packages/tensorflow/contrib/learn/python/learn/datasets/b
ase.py", line 208, in maybe_download
```

```
    temp_file_name, _ = urlretrieve_with_retry(source_url)
```

```
File      "/Users/xuyizhou/anaconda3/lib/python3.6/site-
packages/tensorflow/contrib/learn/python/learn/datasets/b
ase.py", line 165, in wrapped_fn
```

```
    return fn(*args, **kwargs)
```

```
File      "/Users/xuyizhou/anaconda3/lib/python3.6/site-
packages/tensorflow/contrib/learn/python/learn/datasets/b
ase.py", line 190, in urlretrieve_with_retry
```

```
    return urllib.request.urlretrieve(url, filename)
```

```
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.p
y", line 248, in urlretrieve
```

```
    with contextlib.closing(urlopen(url, data)) as fp:
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 223, in urlopen
    return opener.open(url, data, timeout)
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 526, in open
    response = self._open(req, data)
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 544, in _open
    '_open', req)
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 504, in _call_chain
    result = func(*args)
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 1361, in https_open
    context=self._context,
check_hostname=self._check_hostname)
File
"/Users/xuyizhou/anaconda3/lib/python3.6/urllib/request.py", line 1320, in do_open
    raise URLError(err)
URLError: <urlopen error [Errno 60] Operation timed out>
```