WTF Daily Blog

斗大的熊猫

TensorFlow练习15: 中文语音识别

语音识别的应用领域非常广泛，洋文名Speech Recognition。它所要解决的问题是让计算机能够“听懂”人类的语音，将语音中包含的文字信息“提取”出来。

语音识别是前文《聊天机器人》必不可少的一个组件，本帖就使用TensorFlow做一个中文语音识别。

使用的数据集

THCHS30是Dong Wang, Xuewei Zhang, Zhiyong Zhang这几位大神发布的开放语音数据集，可用于开发中文语音识别系统。

为了感谢这几位大神，我是跪在电脑前写的本帖代码。

下载中文语音数据集（5G+）：

```
$ wget http://data.cslt.org/thchs30/zip/wav.tgz
$ wget http://data.cslt.org/thchs30/zip/doc.tgz
$ wget http://data.cslt.org/thchs30/zip/lm.tgz
# 解压
```

```
$ tar xvf wav.tgz
$ tar xvf doc.tgz
$ tar xvf lm.tgz
```

在开始之前，先好好检视一下数据集。

训练

```
import tensorflow as tf  # 0.12
import numpy as np
import os
from collections import Counter
import librosa  # https://github.com/librosa/librosa

# 训练样本路径
wav_path = 'data/wav/train'
label_file = 'data/doc/trans/train.word.txt'

# 获得训练用的wav文件路径列表
def get_wav_files(wav_path=wav_path):
    wav_files = []
    for (dirpath, dirnames, filenames) in os.walk(wav_path):
        for filename in filenames:
            if filename.endswith('.wav') or filename.endswith('.WAV'):
                filename_path = os.sep.join([dirpath, filename])
                if os.stat(filename_path).st_size < 240000:  # 剔除掉一些小文件
                    continue
                wav_files.append(filename_path)
    return wav_files

wav_files = get_wav_files()

# 读取wav文件对应的label
def get_wav_lable(wav_files=wav_files, label_file=label_file):
    labels_dict = {}
    with open(label_file, 'r') as f:
        for label in f:
            label = label.strip('\n')
            label_id = label.split(' ', 1)[0]
```

```
            label_text = label.split(' ', 1)[1]
            labels_dict[label_id] = label_text

    labels = []
    new_wav_files = []
    for wav_file in wav_files:
        wav_id = os.path.basename(wav_file).split('.')[0]
        if wav_id in labels_dict:
            labels.append(labels_dict[wav_id])
            new_wav_files.append(wav_file)

    return new_wav_files, labels

wav_files, labels = get_wav_lable()
print("样本数:", len(wav_files))  # 8911
#print(wav_files[0], labels[0])
# wav/train/A11/A11_0.WAV -> 绿 是 阳春 烟 景 大块 文章 的 底色 四月 的 林 峦 更是 绿 得 鲜活 秀媚 诗意 盎然

# 词汇表(参看练习1和7)
all_words = []
for label in labels:
    all_words += [word for word in label]
counter = Counter(all_words)
count_pairs = sorted(counter.items(), key=lambda x: -x[1])

words, _ = zip(*count_pairs)
words_size = len(words)
print('词汇表大小:', words_size)

word_num_map = dict(zip(words, range(len(words))))
to_num = lambda word: word_num_map.get(word, len(words))
labels_vector = [ list(map(to_num, label)) for label in labels]
#print(wavs_file[0], labels_vector[0])
#wav/train/A11/A11_0.WAV -> [479, 0, 7, 0, 138, 268, 0, 222, 0, 714, 0, 23, 261, 0, 28, 1191, 0, 1, 0, 442,
#print(words[479]) #绿
label_max_len = np.max([len(label) for label in labels_vector])
print('最长句子的字数:', label_max_len)

wav_max_len = 0  # 673
for wav in wav_files:
    wav, sr = librosa.load(wav, mono=True)
    mfcc = np.transpose(librosa.feature.mfcc(wav, sr), [1,0])
```

```
    if len(mfcc) > wav_max_len:
        wav_max_len = len(mfcc)
print("最长的语音:", wav_max_len)

batch_size = 16
n_batch = len(wav_files) // batch_size

# 获得一个batch
pointer = 0
def get_next_batches(batch_size):
    global pointer
    batches_wavs = []
    batches_labels = []
    for i in range(batch_size):
        wav, sr = librosa.load(wav_files[pointer], mono=True)
        mfcc = np.transpose(librosa.feature.mfcc(wav, sr), [1,0])
        batches_wavs.append(mfcc.tolist())
        batches_labels.append(labels_vector[pointer])
        pointer += 1

    # 补零对齐
    for mfcc in batches_wavs:
        while len(mfcc) < wav_max_len:
            mfcc.append([0]*20)
    for label in batches_labels:
        while len(label) < label_max_len:
            label.append(0)
    return batches_wavs, batches_labels

X = tf.placeholder(dtype=tf.float32, shape=[batch_size, None, 20])
sequence_len = tf.reduce_sum(tf.cast(tf.not_equal(tf.reduce_sum(X, reduction_indices=2), 0.), tf.int32), redu
Y = tf.placeholder(dtype=tf.int32, shape=[batch_size, None])

# conv1d_layer
conv1d_index = 0
def conv1d_layer(input_tensor, size, dim, activation, scale, bias):
    global conv1d_index
    with tf.variable_scope('conv1d_' + str(conv1d_index)):
        W = tf.get_variable('W', (size, input_tensor.get_shape().as_list()[-1], dim), dtype=tf.float32, init
        if bias:
            b = tf.get_variable('b', [dim], dtype=tf.float32, initializer=tf.constant_initializer(0))
        out = tf.nn.conv1d(input_tensor, W, stride=1, padding='SAME') + (b if bias else 0)
```

```
        if not bias:
            beta = tf.get_variable('beta', dim, dtype=tf.float32, initializer=tf.constant_initializer(0))
            gamma = tf.get_variable('gamma', dim, dtype=tf.float32, initializer=tf.constant_initializer(1))
            mean_running = tf.get_variable('mean', dim, dtype=tf.float32, initializer=tf.constant_initializer
            variance_running = tf.get_variable('variance', dim, dtype=tf.float32, initializer=tf.constant_in
            mean, variance = tf.nn.moments(out, axes=range(len(out.get_shape()) - 1))
            def update_running_stat():
                decay = 0.99
                update_op = [mean_running.assign(mean_running * decay + mean * (1 - decay)), variance_runnin
                with tf.control_dependencies(update_op):
                    return tf.identity(mean), tf.identity(variance)
                m, v = tf.cond(tf.Variable(False, trainable=False, collections=[tf.GraphKeys.LOCAL_VARIABLES
                out = tf.nn.batch_normalization(out, m, v, beta, gamma, 1e-8)
        if activation == 'tanh':
            out = tf.nn.tanh(out)
        if activation == 'sigmoid':
            out = tf.nn.sigmoid(out)

        conv1d_index += 1
        return out
# aconv1d_layer
aconv1d_index = 0
def aconv1d_layer(input_tensor, size, rate, activation, scale, bias):
    global aconv1d_index
    with tf.variable_scope('aconv1d_' + str(aconv1d_index)):
        shape = input_tensor.get_shape().as_list()
        W = tf.get_variable('W', (1, size, shape[-1], shape[-1]), dtype=tf.float32, initializer=tf.random_un
        if bias:
            b = tf.get_variable('b', [shape[-1]], dtype=tf.float32, initializer=tf.constant_initializer(0))
        out = tf.nn.atrous_conv2d(tf.expand_dims(input_tensor, dim=1), W, rate=rate, padding='SAME')
        out = tf.squeeze(out, [1])
        if not bias:
            beta = tf.get_variable('beta', shape[-1], dtype=tf.float32, initializer=tf.constant_initializer(
            gamma = tf.get_variable('gamma', shape[-1], dtype=tf.float32, initializer=tf.constant_initializer
            mean_running = tf.get_variable('mean', shape[-1], dtype=tf.float32, initializer=tf.constant_init
            variance_running = tf.get_variable('variance', shape[-1], dtype=tf.float32, initializer=tf.const
            mean, variance = tf.nn.moments(out, axes=range(len(out.get_shape()) - 1))
            def update_running_stat():
                decay = 0.99
                update_op = [mean_running.assign(mean_running * decay + mean * (1 - decay)), variance_runnin
                with tf.control_dependencies(update_op):
                    return tf.identity(mean), tf.identity(variance)
```

```
                m, v = tf.cond(tf.Variable(False, trainable=False, collections=[tf.GraphKeys.LOCAL_VARIABLES
                out = tf.nn.batch_normalization(out, m, v, beta, gamma, 1e-8)
        if activation == 'tanh':
            out = tf.nn.tanh(out)
        if activation == 'sigmoid':
            out = tf.nn.sigmoid(out)

        aconv1d_index += 1
        return out
# 定义神经网络
def speech_to_text_network(n_dim=128, n_blocks=3):
    out = conv1d_layer(input_tensor=X, size=1, dim=n_dim, activation='tanh', scale=0.14, bias=False)
    # skip connections
    def residual_block(input_sensor, size, rate):
            conv_filter = aconv1d_layer(input_sensor, size=size, rate=rate, activation='tanh', scale=0.03, b
            conv_gate = aconv1d_layer(input_sensor, size=size, rate=rate,  activation='sigmoid', scale=0.03,
            out = conv_filter * conv_gate
            out = conv1d_layer(out, size=1, dim=n_dim, activation='tanh', scale=0.08, bias=False)
            return out + input_sensor, out
    skip = 0
    for _ in range(n_blocks):
        for r in [1, 2, 4, 8, 16]:
            out, s = residual_block(out, size=7, rate=r)
            skip += s

    logit = conv1d_layer(skip, size=1, dim=skip.get_shape().as_list()[-1], activation='tanh', scale=0.08, bi
    logit = conv1d_layer(logit, size=1, dim=words_size, activation=None, scale=0.04, bias=True)

    return logit

class MaxPropOptimizer(tf.train.Optimizer):
    def __init__(self, learning_rate=0.001, beta2=0.999, use_locking=False, name="MaxProp"):
        super(MaxPropOptimizer, self).__init__(use_locking, name)
        self._lr = learning_rate
        self._beta2 = beta2
        self._lr_t = None
        self._beta2_t = None
    def _prepare(self):
        self._lr_t = tf.convert_to_tensor(self._lr, name="learning_rate")
        self._beta2_t = tf.convert_to_tensor(self._beta2, name="beta2")
    def _create_slots(self, var_list):
        for v in var_list:
```

```
            self._zeros_slot(v, "m", self._name)
    def _apply_dense(self, grad, var):
        lr_t = tf.cast(self._lr_t, var.dtype.base_dtype)
        beta2_t = tf.cast(self._beta2_t, var.dtype.base_dtype)
        if var.dtype.base_dtype == tf.float16:
            eps = 1e-7
        else:
            eps = 1e-8
        m = self.get_slot(var, "m")
        m_t = m.assign(tf.maximum(beta2_t * m + eps, tf.abs(grad)))
        g_t = grad / m_t
        var_update = tf.assign_sub(var, lr_t * g_t)
        return tf.group(*[var_update, m_t])
    def _apply_sparse(self, grad, var):
        return self._apply_dense(grad, var)

def train_speech_to_text_network():
    logit = speech_to_text_network()

    # CTC loss
    indices = tf.where(tf.not_equal(tf.cast(Y, tf.float32), 0.))
    target = tf.SparseTensor(indices=indices, values=tf.gather_nd(Y, indices) - 1, shape=tf.cast(tf.shape(Y)
    loss = tf.nn.ctc_loss(logit, target, sequence_len, time_major=False)
    # optimizer
    lr = tf.Variable(0.001, dtype=tf.float32, trainable=False)
    optimizer = MaxPropOptimizer(learning_rate=lr, beta2=0.99)
    var_list = [t for t in tf.trainable_variables()]
    gradient = optimizer.compute_gradients(loss, var_list=var_list)
    optimizer_op = optimizer.apply_gradients(gradient)

    with tf.Session() as sess:
        sess.run(tf.global_variables_initializer())

        saver = tf.train.Saver(tf.global_variables())

        for epoch in range(16):
            sess.run(tf.assign(lr, 0.001 * (0.97 ** epoch)))

            global pointer
            pointer = 0
            for batch in range(n_batch):
                batches_wavs, batches_labels = get_next_batches(batch_size)
```

```
                train_loss, _ = sess.run([loss, optimizer_op], feed_dict={X: batches_wavs, Y: batches_labels}
                print(epoch, batch, train_loss)
            if epoch % 5 == 0:
                saver.save(sess, 'speech.module', global_step=epoch)

# 训练
train_speech_to_text_network()

# 语音识别
# 把batch_size改为1
def speech_to_text(wav_file):
    wav, sr = librosa.load(wav_file, mono=True)
    mfcc = np.transpose(np.expand_dims(librosa.feature.mfcc(wav, sr), axis=0), [0,2,1])

    logit = speech_to_text_network()

    saver = tf.train.Saver()
    with tf.Session() as sess:
        saver.restore(sess, tf.train.latest_checkpoint('.'))

        decoded = tf.transpose(logit, perm=[1, 0, 2])
        decoded, _ = tf.nn.ctc_beam_search_decoder(decoded, sequence_len, merge_repeated=False)
        predict = tf.sparse_to_dense(decoded[0].indices, decoded[0].shape, decoded[0].values) + 1
        output = sess.run(decoded, feed_dict={X: mfcc})
        #print(output)
```

后续：从麦克风获得语音输入，使用上面的模型进行识别。

相关资源：

- TensorFlow练习8: 基于RNN生成音乐
- Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning
- 深度学习大牛Andrew Ng：Speech Recognition and Beyond
- https://github.com/kaldi-asr/kaldi
- http://cmusphinx.sourceforge.net
- https://pypi.python.org/pypi/SpeechRecognition

相关文章

Ubuntu 16.04 安装 Tensorflow(GPU支持)

使用Python实现神经网络

TensorFlow练习1: 对评论进行分类

TensorFlow练习2: 对评论进行分类

TensorFlow练习4: CNN, Convolutional Neural Networks...

2016年12月10日 wtf ML、coding TensorFlow、中文语音识别

《TensorFlow练习15: 中文语音识别》有20个想法

zerozzl

2017年6月8日 下午3:27

大神你好，解码的时候，得到的logit是有数值的，但是解码的结果，是一个空值，请问是哪里出错了吗？
[SparseTensorValue(indices=array([], shape=(0, 2), dtype=int64), values=array([], dtype=int64), dense_shape=array([1, 0]))]

hc

2017年6月9日 上午10:37

我和你同样的问题，我的words大小是2269，python2.7,tensorflow0.12

catmonkey

2017年4月20日 下午12:31

训练结束后，预测的时候报错，可否指点下？
Failed precondition: Attempting to use uninitialized value conv1d_18/W

connor

2017年4月18日 下午4:43

我自己笔记本连着训练了几天，还没训练完，结果笔记本死机了，估计前面那些是白训练了。后续还有的出吗？

李勇

2017年3月5日 下午7:47

all_words += [word for word in label] 这里是不有bug label需要split

hc

2017年6月9日 上午10:37

python版本的原因，你用的是python2.7吧

---

acans

2017年1月17日 下午6:02

后续：从麦克风获得语音输入，使用上面的模型进行识别。
这个后续什么时候能出？
楼主能给个如何使用这个模型来输出的例子吗?

---

wtf

2017年1月17日 下午6:11

后续纯粹唬人，我这水平太水，不要被我坑了，速速弃坑

---

hc

2017年6月9日 上午10:38

大神你好，解码的时候，得到的logit是有数值的，但是解码的结果，是一个空值，请问是哪里出错了吗？
[SparseTensorValue(indices=array([], shape=(0, 2), dtype=int64), values=array([], dtype=int64), dense_shape=array([1, 0]))]

---

zhanglihai

2017年1月10日 上午10:10

遇到了wav.zip 文件下载失败的问题。尝试5次每次都在36%失败。

ten2net

2017年1月10日 下午12:31

用迅雷下载后传到服务器上解压。我就是这么做的

ten2net

2017年1月9日 下午4:27

我用wav/train/A6/A6_111.wav来预测。结果返回……

...
values=array([0, 0, 38, 0, 5, 38, 0, 5, 38, 0, 5, 38, 0, 5, 38, 0, 5,
9, 0, 5, 9, 0, 5, 15, 0, 5, 9, 0, 16, 0, 6, 0, 5, 6,
0, 5, 0, 5, 36, 0, 14, 15, 0, 14, 15, 0, 5, 9, 0, 5, 35,
0, 5, 0, 5, 6, 0, 6, 0, 5, 9, 0, 9, 0, 0, 6, 0, 6,
15, 0, 6, 15, 0, 5, 15, 0, 14, 9, 0, 14, 15, 0, 0]
...

使用如下代码转换，结果却是乱码，请指教哪儿不对：
python
msg=''.join([words[n] for n in output[0][1]])
print msg

wtf

2017年1月9日 下午5:48

调整一下坐姿，再来一次

---

zjun

2017年6月5日 下午1:09

为啥我返回的是空啊。。。您遇到过这种情况么，

---

meng_xi

2017年1月18日 下午9:14

能大概解释下吗，请问这个是什么意思呢？您是怎么做的呢？

---

cdjauto

2017年3月14日 上午11:45

我识别的时候都是得到这个结果：
[SparseTensorValue(indices=array([[0, 0],
[0, 1],
[0, 2],
[0, 3],
[0, 4],
[0, 5],
[0, 6],
[0, 7],
[0, 8],
[0, 9]]), values=array([7113, 7113, 7113, 7113, 7113, 7113, 7113, 7113, 7113, 7113]), shape=array([1, 10]))]
是我打开的姿势不对吗？求指教

garbo

2017年7月7日 上午9:38

请问这套代码训练完了之后，做预测应该如何实现？

ten2net

2017年1月9日 下午4:16

期待博主的后续中……

wtf

2017年1月9日 下午4:25

何年何月才有后续…

ervin

2016年12月18日 上午10:45

楼主你好！非常感谢你分享的实践经验！我在调试运行你的代码，但是发现训练时间实在太长，需要几天的时间。
1.请问楼主你是用什么硬件资源训练的网络？花费的时间怎样？
2.还有，我的笔记本配置有限，你可否方便把你训练好的网络发给我研究一下呢？真诚感谢！