CSDN (http://www.csdn.net)

目录

tf15: 中文语音识别

喜欢 收藏 评论 分享

原创 2016年12月25日 14:22:54 9883 46 4

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 (http://data.cslt.org/thchs30/standalone.html)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+）：

```
[python]
$ 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

```
[python]
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'):
```

目录
喜欢
收藏
评论
分享

是AI就躲个飞机-纯Python实现人工智能…
Ubuntu查看当前IP地址 (/lllody/article/d…
02 The TensorFlow Way（1）(/u0143…

返回顶部

```
                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, 199, 0, 72, 3
#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.), t
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, initializer=tf.random_uniform_initializer(minval=-
scale, maxval=scale))
        if bias:
            b = tf.get_variable('b', [dim], dtype=tf.float32, initializer=tf.constant_initializ
        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_initi
            gamma = tf.get_variable('gamma', dim, dtype=tf.float32, initializer=tf.constant_ini
            mean_running = tf.get_variable('mean', dim, dtype=tf.float32, initializer=tf.consta
            variance_running = tf.get_variable('variance', dim, dtype=tf.float32, initializer=t
            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)), va
                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]), update_running_stat, lambda: (mean_running, variance_running))
                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
scale, maxval=scale))
        if bias:
            b = tf.get_variable('b', [shape[-1]], dtype=tf.float32, initializer=tf.constant_ini
        out = tf.nn.atrous_conv2d(tf.expand_dims(input_tensor, dim=1), W, rate=rate, padding='S
        out = tf.squeeze(out, [1])
        if not bias:
            beta = tf.get_variable('beta', shape[-1], dtype=tf.float32, initializer=tf.constant
            gamma = tf.get_variable('gamma', shape[-1], dtype=tf.float32, initializer=tf.consta
            mean_running = tf.get_variable('mean', shape[-1], dtype=tf.float32, initializer=tf.
            variance_running = tf.get_variable('variance', shape[-1], dtype=tf.float32, initial
            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)), va
                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]), update_running_stat, lambda: (mean_running, variance_running))
                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=F
    # skip connections
    def residual_block(input_sensor, size, rate):
            conv_filter = aconv1d_layer(input_sensor, size=size, rate=rate, activation='tanh',
            conv_gate = aconv1d_layer(input_sensor, size=size, rate=rate,  activation='sigmoid'
```

```
            out = conv_filter * conv_gate
            out = conv1d_layer(out, size=1, dim=n_dim, activation='tanh', scale=0.08, bias=Fals
            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, bias=False)
    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):
```

目录
喜欢
收藏
评论
分享
返回顶部

```
        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.cas
    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) +
        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 (http://blog.topspeedsnail.com/archives/10508)

深度学习大牛Andrew Ng：Speech Recognition and Beyond (https://www.youtube.com/watch?v=LFDU2GX4AqM)

https://github.com/kaldi-asr/kaldi (https://github.com/kaldi-asr/kaldi)

http://cmusphinx.sourceforge.net (http://cmusphinx.sourceforge.net/)

https://pypi.python.org/pypi/SpeechRecognition (https://pypi.python.org/pypi/SpeechRecognition)

标签：Tensorflow (http://so.csdn.net/so/search/s.do?q=Tensorflow&t=blog) /

qq_15144693 (/qq_15144693) 2017-09-12 20:39 24楼

(/qq_15144693)我和其他人一样，使用ctc解码，返回值为空，请问你知道怎么解决吗

回复

查看 46 条热评

相关文章推荐

是AI就躲个飞机-纯Python实现人工智能 (/u014365862/article/details/54380422)

很久以前微信流行过一个小游戏：打飞机，这个游戏简单又无聊。在2017年来临之际，我就实现一个超级弱智的人工智能（AI），这货可以躲避从屏幕上方飞来的飞机。本帖只使用纯Python实现，不依赖任何高级库...

u014365862 (http://blog.csdn.net/u014365862) 2017-01-12 18:08 12591

Ubuntu查看当前IP地址 (/lllody/article/details/6640470)

ifconfig eth0 |awk '/inet/ {split($2,x,":");print x[2]}'截图： 网络 根据IP查网卡地址 arping IP地址根据IP查电脑名 nmblook...

lllody (http://blog.csdn.net/lllody) 2011-07-28 11:38 3734

精选：深入理解 Docker 内部原理及网络配置 (http://edu.csdn.net/huiyiCourse/detail/563?utm_source=blog10)

网络绝对是任何系统的核心，对于容器而言也是如此。Docker 作为目前最火的轻量级容器技术，有很多令人称道的功能，如 Docker 的镜像管理。然而，Docker的网络一直以来都比较薄弱，所以我们有必要深入了解Docker的网络知识，以满足更高的网络需求。

02 The TensorFlow Way（1）(/u014365862/article/details/70884624)

The TensorFlow Way Introduction： 现在我们介绍了TensorFlow如何创建张量，使用变量和占位符，我们将介绍如何在计算图中对这些对象采取处理。 从...

u014365862 (http://blog.csdn.net/u014365862) 2017-04-28 10:09 305

tf1: nn实现评论分类 (/u014365862/article/details/53868418)

原文链接： http://blog.topspeedsnail.com/archives/tag/tensorflow TensorFlow是谷歌2015年开源的一个深度学习库，到现在正好一年。和T...

u014365862 (http://blog.csdn.net/u014365862) 2016-12-25 09:54 2498

01 TensorFlow入门（2）(/u014365862/article/details/70849334)

Working with Matrices： 了解TensorFlow如何使用矩阵对于通过计算图理解数据流非常重要。 Getting ready： ...

u014365862 (http://blog.csdn.net/u014365862) 2017-04-27 18:12 366

tf14: 黑白图像上色 (/u014365862/article/details/53869682)

网上有很多使用PS给黑白老照片上色的教程，下图是P上颜色的爱因斯坦： 于是，有大牛使用深度学习制作了一个自动给黑白图像上色的模型，非常非常的牛叉。 colornet Automatic ...

u014365862 (http://blog.csdn.net/u014365862) 2016-12-25 14:20 2903

tf2: nn和cnn实现评论分类 (/u014365862/article/details/53868422)

原文链接： http://blog.topspeedsnail.com/archives/tag/tensorflow 本帖是前一贴的补充： 使用大数据，了解怎么处理数据不能一次全部加载到内存的...

u014365862 (http://blog.csdn.net/u014365862) 2016-12-25 09:55 1874

01 TensorFlow入门（1）(/u014365862/article/details/70837638)

tensorflow_cookbook--第1章 TensorFlow入门 Google的TensorFlow引擎具有独特的解决问题的方法。 这种独特的方式允许我们非常有效地解决机器...

u014365862 (http://blog.csdn.net/u014365862) 2017-04-27 09:51 579

修改HTK代码，让其支持中文 (/xiaoding133/article/details/8491341)

利用HTK工具包进行语音识别建模时，遇到任务语法中存在中文时候，无法生成对应的底层网络，这样就需要对HTK源码的部分内容进行修改，以下是我对HTK源码HParse及HVite部分内容改动记录，希望对有...


xiaoding133 (http://blog.csdn.net/xiaoding133) 2013-01-10 20:25 3637

语音识别的例子 (/liuyuehui110/article/details/72636266)

在.NET4.0中，我可以借助System.Speech组件让电脑来识别我们的声音。 以上，当我说"name"，显示"Darren"，我说"age",显示"永远21"。如何做呢...


liuyuehui110 (http://blog.csdn.net/liuyuehui110) 2017-05-23 07:12 253


目录


喜欢


收藏


评论


分享

tf15: 中文语音识别 (/u014365862/article/details/53869701)

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...


u014365862 (http://blog.csdn.net/u014365862) 2016-12-25 14:22 9884

TensorFlow 中文语音识别 (/sinat_30665603/article/details/74897891)

本文转载自 http://blog.topspeedsnail.com/archives/10696 数据集下载参见该文。 其中下面的代码进行了一些小小的调整。其中包含缩进、版本方面(作...


sinat_30665603 (http://blog.csdn.net/sinat_30665603) 2017-07-09 18:09 745

DTW 动态时间规整 (/u010554204/article/details/26621185)

HMM学习笔记_1(从一个实例中学习DTW算法) DTW为(Dynamic Time Warping,动态时间归准)的简称。应用很广，主要是在模板匹配中，比如说用在孤立词语音识...


u010554204 (http://blog.csdn.net/u010554204) 2014-05-22 20:53 472


返回顶部

语音基础知识 (/weixin_37355348/article/details/75127427)

语音识别原理： 语音识别的最终目的是让机器听懂人的语言。 语音信号通过麦克风采集，经过采样和 A/D 转换后由模拟信号转变为数字信号。然后对语音...


weixin_37355348 (http://blog.csdn.net/weixin_37355348) 2017-07-14 18:29 71

目录
喜欢
收藏
评论
分享

提高开发人员工作效率的7个技巧 (/stefan520/article/details/13622603)

谁不希望有更多的时间来解决那些费神的复杂任务？利用一些节省时间的技巧来优化工作流程有助于在更短的时间内完成更多的工作。本文总结了帮助开发人员提供工作效率的7条技巧，供大家参考。 使用语音识别 如果...


stefan520 (http://blog.csdn.net/stefan520) 2013-10-30 10:23 439

1.简谈语音识别中的WFTS (/qq_16949707/article/details/53197698)

用WFST来表征ASR中的模型（HCLG），可以更方便的对这些模型进行融合和优化，于是可以作为一个简单而灵活的ASR的解码器（simple and flexible ASR decoder desig...


qq_16949707 (http://blog.csdn.net/qq_16949707) 2016-11-17 13:47 787

从声学模型算法总结 2016 年语音识别的重大进步 (/charleslei/article/details/58142000)

从声学模型算法总结 2016 年语音识别的重大进步


charleslei (http://blog.csdn.net/charleslei) 2017-02-27 16:57 2790

5-基于决策树的状态绑定 (/victoryaoyu/article/details/70308584)

本文会对GMM-HMM模型的建模单元Context-dependent model (CD)的优化方法——基于决策树的状态绑定，进行一个原理介绍。Outline：Context-dependent m...

返回顶部