Kaldi 乱弹--语音数据增强

原创：静默 Kaldi 学习 前天

众所周知，数据增强无外乎两个目的：

- 增强训练数据量，提高模型的泛化能力
- 增加噪声数据，提高模型的鲁棒性

如果大家接触过深度学习，自然接触过数字识别，在数字识别中存在数据增强算法，本文不会对图强的数字增强算法进行说明。同样，在语音识别中也存在类似的增强算法。

本文主要是通过 Kaldi 内相关语音数据增强的方法来进行说明。

如何增加训练数据量？

在 Kaldi 中 的增强训练数据量的做法是，通过对训练语音进行加速和减速进行语音数据增强。从而提升训练数据量。该增强数据的脚本主要存在于：egs/wsj/s5/utils/data/perturb_data_dir_speed_3way.sh

该脚本的使用方法：

该脚本需要两个参数，第一个是原训练语音目录，另一个参数为增强后的语音目录。这里所说的原训练语音目录是指经过数据处理之后存放于 data 目录中的训练数据目录。而另一个增强后的语音目录指的是之后需要生成的目录。

这里会对数据进行增速和减速，其中默认的增减速倍数为0.9、1.1 和 1.0，如果各位需要跟多倍数的加减速，可以针对该脚本进行修改。

如何增加噪声数据，提高模型的鲁棒性？

这里需要介绍一个网站 OpenSLR 网址：http://www.openslr.org/resources.php，这里主要是开源语音数据。而接下来需要介绍的噪声数据也来自该网址。分别是 17 和 28 。该方法需要自己编写组合才能有效。脚本编写如下：

```
1
2 . ./cmd.sh
3 . ./path.sh
```

```
4  set -e
5  mfccdir=`pwd`/mfcc
6  vaddir=`pwd`/mfcc
7
8
9  stage=1
10 stopstage=1
11
12 noise_data='/data/noise_data/RIRS_NOISES'
13 music_data='/data/noise_data/music'
14 # add RIRs, simulated RIRs, isotropic noises and point-source noises
15 if [ $stage -le 0 -a $stopstage -ge 0 ]; then
16   frame_shift=0.01
17   # get audio duration
18   awk -v frame_shift=$frame_shift '{print $1, $2*frame_shift;}' data/train/ut
19
20   # Make a version with reverberated speech
21   rvb_opts=()
22   rvb_opts+=(--rir-set-parameters "0.5, RIRS_NOISES/simulated_rirs/smallroom/r
23   rvb_opts+=(--rir-set-parameters "0.5, RIRS_NOISES/simulated_rirs/mediumroom/
24
25   # Make a reverberated version of the SWBD+SRE list.  Note that we don't add
26   # additive noise here.
27   python3 steps/data/reverberate_data_dir.py \
28     "${rvb_opts[@]}" \
29     --speech-rvb-probability 1 \
30     --pointsource-noise-addition-probability 0 \
31     --isotropic-noise-addition-probability 0 \
32     --num-replications 1 \
33     --source-sampling-rate 16000 \
34     data/train data/train_reverb
35   #cp data/dev/vad.scp data/dev_reverb/
36   utils/copy_data_dir.sh --utt-suffix "-reverb" data/train_reverb data/train_r
37   rm -rf data/train_reverb
38   mv data/train_reverb.new data/train_reverb
39 fi
40
41   # Prepare the MUSAN corpus, which consists of music, speech, and noise
42   # suitable for augmentation.
43 if [ $stage -le 1 -a $stopstage -ge 1 ]; then
```

```
  local/make_musan.sh $music_data data

  # Get the duration of the MUSAN recordings.  This will be used by the
  # script augment_data_dir.py.
  for name in speech noise music; do
    utils/data/get_utt2dur.sh data/musan_${name}
    mv data/musan_${name}/utt2dur data/musan_${name}/reco2dur
  done

  # Augment with musan_noise
  python3 steps/data/augment_data_dir.py --utt-suffix "noise" --fg-interval 1
  # Augment with musan_music
  python3 steps/data/augment_data_dir.py --utt-suffix "music" --bg-snrs "15:10
  # Augment with musan_speech
  python3 steps/data/augment_data_dir.py --utt-suffix "babble" --bg-snrs "20:1

  # Combine reverb, noise, music, and babble into one directory.
  utils/combine_data.sh data/train_combine data/train_reverb data/train_noise


  # Make MFCCs for the augmented data.  Note that we do not compute a new
  # vad.scp file here.  Instead, we use the vad.scp from the clean version of
  # the list.
  steps/make_mfcc.sh --write-utt2num-frames true --mfcc-config conf/mfcc_hires
    data/train_combine exp/make_mfcc $mfccdir
fi
```

下一篇文章会详细介绍该算法。
欢迎大家推荐身边对语音识别、Kaldi 感兴趣的朋友添加关注此公众号。

阅读原文