1

專題研究 Week 1 Introduction

Prof. Lin-Shan Lee TA: Chung-Ming Chien



Project Introduction
 Linux and Bash Introduction
 Feature Extraction
 Homework





◆ 目的:透過建立一個基本的大字彙語音辨識系統,讓同學對語音辨識有具體的了解,並且以此作為進一步研究各項進階技術的基礎。





- 5
- Conventional ASR (Automatic Speech Recognition) system:



Deep learning based ASR system



- Conventional ASR system
 - Widely used in commercial system
- Deep learning based ASR system
 - Widely studied in recent years

Both will be implemented in this project with Kaldi toolkit

Schedule

7

階

段

	Week	Progress	Report Group
♦	1	Introduction + Linux intro+ Feature extraction	
第	2	Acoustic model training : monophone & triphone	
	3	Language model training + Decoding	А
階 段	4	Live demo	
	5	Deep Neural Network	А
	6	Progress Report	В
¥	7	Progress Report	А
▼ 第		••••	



Conventional ASR (Automatic Speech Recognition) system:



Deep learning based ASR system

Week 5

How to do recognition?

- 9
- How to map speech O to a word sequence W ?

$$\widehat{W} = \arg \max_{W} P(W|O)$$

= $\arg \max_{W} \frac{P(O|W)P(W)}{P(O)}$
= $\arg \max_{W} P(O|W)P(W)$

- P(O | W): acoustic model
- P(W): language model

Language model P(W)

•
$$W = w_1, w_2, w_3, ..., w_n$$

$$P(W) = P(W_1)P(W_2|W_1) \prod_{i=3}^{n} P(W_i|W_{i-2}, W_{i-1})$$

Language model examples

11

	\2-grams:	
	-2.465383	<s> 不</s>
	-2.465383	<s> 台灣</s>
	-2.465383	<s> 有六</s>
	-2.465383	<s> 但是</s>
	-2.465383	<s> 其中</s>
log Proh	-2.465383	<s> 淳樸</s>
IUG FIUD	-2.465383	<s> 這</s>
	-2.465383	<s> 愛</s>
	-2.465383	<s> 暴</s>
	-2.465383	<s> 雖然</s>
	-1.465383	一 一處 偏僻
	-1.465383	一 一萬 兩千
	-1.942504	了是
	-1.942504	
	-1.942504	了稀釋
	-1.465383	人 看
	-1.766413	也呼籲
	-1.766413	也 得
	-1.465383	
	-1.465383	大學的最新
	-1.465383	小心 才能
	-1.465383	ビ ビ 選 達到
	-1.465383	<u>才能</u> ,有效
	-1.766413	<u> </u>
	-1.766413	
	-1.465383	
	-1.465383	こ 2外_ 柿釋
	-1.465383	六 白人

Acoustic Model P(O | W)

12

Model of a phone



Lexicon

CH_p CH_e CH_i CH_# CH_y CH_p CH_e CH_i CH_b CH_a CH_n# CH_p CH_e CH_i CH_t CH_o CH_N# CH_p CH_e CH_i CH_k CH_@ CH_p CH_e CH_i CH_k CH_@ CH_p CH_e CH_i CH_k CH_a CH_u CH_p CH_e CH_i CH_dzi CH_i CH_p CH_e CH_i CH_tsc CH_0 CH_n# CH_p CH_e CH_i CH_tsi CH_i CH_E CH_n# CH_p CH_e CH_i CH_tsc CH_a CH_N# CH_p CH_e CH_i CH_dz CH_u CH_e CH_i 帶 СН_р СН_е СН_і СН_d СН_а СН_і CH_p CH_e CH_i CH_b CH_e CH_i CH_p CH_e CH_i CH_f CH_a 配方 CH_p CH_e CH_i CH_f CH_a CH_N# CH_p CH_e CH_i CH_d CH_u CH_e CH_i CH_p_CH_e_CH_i_CH_t_CH_a_CH_u ˈCH_p_CH_e_CH_i_CH_h_CH_@ ˈCH_p CH_e CH_i CH_dzi CH_i 記件 CH_p CH_e CH_i CH_dzi CH_i CH_E CH_n# CH_p CH_e CH_i CH_dzi CH_y CH_E CH_p CH_e CH_i CH_dzi CH_i CH_a CH_u CH_p CH_e CH_i CH_si CH_i CH_a CH_N# CH_p CH_e CH_i CH_dzc CH_Uc CH_p CH_e CH_i CH_sc CH_o CH_u CH_p CH_e CH_i CH_sc CH_a CH_N# 已所 CH_p CH_e CH_i CH_s CH_u CH_o 記額 CH_p CH_e CH_i CH_# CH_@ CH_p CH_e CH_i CH_# CH_o CH_u



- 5
- Conventional ASR (Automatic Speech Recognition) system:



Deep learning based ASR system



Vim

• 如何建立文件:

- vim hello.txt
- ◆ 進去後, 輸入 "i" 即可進入編輯模式
 - 此時,輸入任何你想要打的
- ◆ 此時,按下ESC即可回復一般模式,此時可以:
 - ■輸入" /想搜尋的字 "
 - ■輸入":w"即可存檔
 - ■輸入":wq"即可存檔+離開



- 簡單講一下,避免因為斷線而程式跑到一半就失敗了, 大家可以使用screen,簡單使用法如下:
 1. 一登入後打"screen",就進入了screen使用模式,用法都相同
 2. 如果想要關掉此screen也是用"exit"
 3. 如果還有程式在跑沒有想關掉他,但是想要跳出, 按"Ctrl + a" + "d"離開screen模式(此時登出並關機程式也不會斷掉)
 4. 下次登入時,打"screen -r"就可以跳回之前沒關掉的screen唷~
 5. 打" screen -r" 也許會有很多個未關的screen,輸入你要的 screen id 即可(越大的越新)
- ◆ 這樣就算關掉電腦,工作仍可以進行!!!
- ◆ 也可以用tmux[→] tmux像是有更多功能的screen

Linux Shell Script Basics

- 18
- echo "Hello" (print "hello" on the screen)
- ♦ a=ABC (assign ABC to a)
- ◆ echo \$a 用變數)
- (will print ABC on the screen, \$: 取
- b=\$a.log (assign ABC.log to b)
- cat \$b > testfile (write "ABC.log" to testfile)
- ◆ 指令 -h (will output the help information)

Bash Example

1	#!/bin/bash	1	#!/bin/bash
2		2	
3	count=99	3	i=0
4	if [\$count 	4	while [\$i <i>—lt</i> 10]
5	then	5	do
6	echo "Count is 100"	6	echo \$i
7	elif [\$count -<i>gt</i> 100]	7	i=\$((\$i+1))
8	then	8	
9	echo "Count is greater than 100"	9	
10	else	10	<pre>for ((i=1; i<=10; i=i+1))</pre>
11	echo "Count is less than 100"	11	do
12	fi	12	echo \$i
12		13	done

- [condition] uses 'test' to check. Ex. test -e ~/tmp; echo \$?
- File [-e filename]

\$?: 取用上一個指令的回傳值

- □ -e 該「檔名」是否存在?
- □ -f 該「檔名」是否存在且為檔案(file)?
- □ -d 該「檔名」是否存在且為目錄(directory)?
- Number [n1 -eq n2]
 - \Box -eq equal (n1==n2)
 - \square -ne not equal (n1!=n2)
 - \Box -gt greater than (n1>n2)
 - □ -It less than (n1<n2)
 - \Box -ge greater or equal (n1>=n2)
 - -le less than or equal (n1 < = n2)

♦ SPACE COUNTS!!!!

- Logic
 - □ -a and
 - □ -0 Or
 - □ ! negation
- ["\$yn" == "Y" -o "\$yn" == "y"]
- ["\$yn" == "Y"] || ["\$yn" == "y"]
- Don't forget the space and the double quote!!!!

- `operation
 _echo `ls`
 _my_date=`date`
 _echo \$my_date
- && || ; operation
 - echo hello || echo no~
 - echo hello && echo no~
 - [-f tmp] && cat tmp || echo "file not found"
 - [-f tmp]; cat tmp; echo "file not found"
- Some useful commands.

grep, sed, touch, awk, In

- Pipeline
- program1 | program2 | program3
- echo "hello" | tee log
- More information about pipeline: <u>http://www.gnu.org/software/bash/manual/html_node/Pipelines.html</u>





Input / output for bash:

cmd > logfile #將 stdout 導入logfile, stderr 印於螢幕 cmd > logfile 2>&1 #將stdout、stderr 全部導到 logfile cmd <inputfile 2>errorfile | grep stdoutfile

 More Information about bash input/output: <u>http://tldp.org/LDP/Bash-Beginners-</u> <u>Guide/html/sect_08_02.html</u>

25 Feature Extraction

02.extract.feat.sh



Feature Extraction - MFCC



MFCC (Mel-frequency cepstral coefficients)



Extract Feature (02.extract.feat.sh)



echo "Extracting dev set" target=dev Log=\$path/\$target.extract.log compute-mfcc-feats --verbose=2 \$options scp:material/\$target.wav.scp ark,t,scp:\$path/\$target.13.ark,\$path/\$target.13.scp 2> \$log # TODO: # 1. use add-deltas to add delta features to original mfcc feature set # 1. use add-deltas to add delta features to original mfcc feature set

2. use compute-cmvn-stats & apply-cmvn to compute cmvn statistics and apply cmvn with global means and variatnces to previous 39-dim mfcc feature set

echo "Extracting test set" Testing Set target=test

log=\$path/\$target.extract.log

compute-mfcc-feats --verbose=2 \$options scp:material/\$target.wav.scp ark,t,scp:\$path/\$target.13.ark,\$path/\$target.13.scp 2> \$log
TODO:

1. use add-deltas to add delta features to original mfcc feature set

2. use compute-cmvn-stats & apply-cmvn to compute cmvn statistics and apply cmvn with global means and variatnces to previous 39-dim mfcc feature set

Kaldi rspecifier & wspecifier format

- 29
- ark:<ark file> 眾多小檔案的檔案庫,可能是 wav檔、mfcc檔、statistics的集合
- ◆ scp:<scp file> 一群檔案的位置表,可能指向 個別檔案(如我們的material/train.wav.scp), 也可以指向ark檔中的位置
- ◆ ark,t:<ark file> 輸出文字檔案的ark,當輸入
 時,t無作用;不加,t,預設輸出二進位格式
- ark,scp:<ark file>,<scp file> 同時輸出ark檔 和scp檔

Extract Feature (02.extract.feat.sh)

- compute-mfcc-feats
- add-deltas
- compute-cmvn-stats
- apply-cmvn

MFCC – Add delta

add-deltas

- Deltas and Delta-Deltas
- ◆ 將MFCC的△以及△△(意近一次微分與二次微分) 加入參數中,使得總維度變成39維
- ◆ Usage :

Add deltas (typically to raw mfcc or plp features Usage: add-deltas [options] in-rspecifier out-wspecifier Options: --delta-order : Order of delta computation (int, default = 2) --delta-window : Parameter controlling window for delta computation (actual window size for each delta order is 1 + 2* elta-window-size) (int, default = 2) --truncate : If nonzero, first truncate features to this dimension. (int, default = 0)

MFCC – CMVN

- ♦ CMVN :
- Cepstral Mean and Variance Normalization

MFCC – CMVN

compute-cmvn-stats

♦ Usage :

Compute cepstral mean and variance normalization statistics If wspecifier provided: per-utterance by default, or per-speaker if spk2utt option provided; if wxfilename: global

Usage: compute-cmvn-stats [options] feats-rspecifier (stats-wspecifier|stats-wxfilename)

Options:

--binarv

: write in binary mode (applies only to global CMN/CVN) (bool, default = true)

- : rspecifier for speaker to utterance-list map (string, default = "")
- --spk2utt --weights
 - : rspecifier for a vector of floats for each utterance, that's a per-frame weight. (string, default = ""
 - apply-cmvn

♦ Usage :

Apply cepstral mean and (optionally) variance normalization Per-utterance by default, or per-speaker if utt2spk option provided Usage: apply-cmvn [options] (cmvn-stats-rspecifier|cmvn-stats-rxfilename) feats-rspecifier feats-wspecifier

Options:

norm-means	: You can set this to false to turn off mean normalization.	Note, the same can be achieved by using 'fa
e' CMVN stats; see thefak	e option to compute_cmvn_stats.sh (bool, default = true)	
norm-vars	: If true, normalize variances. (bool, default = false)	
utt2spk	: rspecifier for utterance to speaker map (string, default =	"")

Hint (Important!!)

- compute-mfcc-feats
 - output為ark:\$path/\$target.13.ark
- add-deltas [input] [output]
 - > [input] = ark:\$path/\$target.13.ark
 - [output] = x
- compute-cmvn-stats [input] [comput_result]
 - > [input] = x
- apply-cmvn [comput_result] [input] [output]
 - Foutput] MUST BE

ark,t,scp:\$path/\$target.39.cmvn.ark,\$path/\$target.39.cmvn.scp

rm -f [output] [comput_result]

MFCC – CMVN

- ♦ CMVN :
- Cepstral Mean and Variance Normalization



Linux, background knowledge 01.format.sh, 02.extract.feat.sh

Homework

- 如果你沒有操作 Linux 系統的經驗,請事 先預習 Linux 系統的指令。
 鳥哥的Linux 私房菜
 - 第七章Linux 檔案與目錄管理
 <u>http://linux.vbird.org/linux_basic/0220filemanager.</u>
 <u>php</u>
 - □ 第十章vim 程式編輯器 <u>http://linux.vbird.org/linux_basic/0310vi.php</u>

Homework (optional)

- 閱讀:
- 使用加權有限狀態轉換器的基於混合詞與次詞以 文字及語音指令偵測口語詞彙" – 第三章 <u>https://www.dropbox.com/s/dsaqh6xa9dp3dzw/wfst_thesis.pdf</u>
- Kaldi documentation :
 <u>http://kaldi-asr.org/doc/tools.html</u>

Login Workstation

- By pietty/putty/Xshell ssh 140.112.21.80 port 22
- By terminal ssh -p 22 username@140.112.21.80



- ▶ 將壓縮檔複製至自己的家目錄底下 cp /share/proj1.ASTMIC.subset.tar.gz ~/.
- ◆ 解壓縮

tar -zxvf proj1.ASTMIC.subset.tar.gz

To Do: Feature Extraction

- Step 1: Execute the following command:
 - script/01.format.sh | tee log/01.format.log
 - script/02.extract.feat.sh | tee log/02.extract.feat.sh.log
- ♦ Step 2:
 - Add-delta
 - CMVN
- Observe the output and report

Hint (Important!!)

- compute-mfcc-feats
 - output為ark:\$path/\$target.13.ark
- add-deltas [input] [output]
 - > [input] = ark:\$path/\$target.13.ark
 - [output] = x
- compute-cmvn-stats [input] [comput_result]
 - > [input] = x
- apply-cmvn [comput_result] [input] [output]
 - Foutput] MUST BE

ark,t,scp:\$path/\$target.39.cmvn.ark,\$path/\$target.39.cmvn.scp

rm -f [output] [comput_result]

工作站注意事項

- 請避免在程式中重複暴力的搜尋外網或抓取資料,這類 的行為如果被計中偵測到,會將ip給ban,造成大家無 法連進工作站。
- 如果需要train的corpus佔用空間需要超過50G以上, 麻煩請寄信給我,以控制專題工作站的空間使用量。
 因工作站運算資源有限,請避免使用工作站train一些個 人作業等,而讓資源留給大家使用在專題研究上。
 本次project中,Week 3 & Week 5 的實驗需要的大量 運算資源和時間,請大家儘早開始,免得積到最後一兩 天,大家的程式會因運算資源有限,而造成全部卡住, 大家都無法進行實驗。

工作站注意事項

- 5. 有為大家裝不同版本的cuda library,大家如果在某些 檔案需要使用各版本的cuda library,請自行加進path 中,如果所需要的cuda版本沒有,可以寫信請我幫忙 裝。
- 6. 第二階段專題的時候建議大家使用virtual environment, Ex: virtualenv, conda等,也可以使用 pip--user將需要的package放在local端
 7. 請不要在工作站跑ipython之類互動式的程式,會吃掉 大家的資源,請直接跑python檔。

其他注意事項

- Problems about the project:
 Facebook Group : <u>數位語音專題</u>
 DSP Website:
 - - http://speech.ee.ntu.edu.tw/courses.html Week 1 TA: 简仲明 r08922080@ntu.edu.tw
- Problems about the workstation:

 - Facebook Group:數位語音專題 Workstation TA: 簡仲明 <u>r08922080@ntu.edu.tw</u>

其他注意事項

- You are encouraged to post your problems on the Facebook group. Your problems may be others' too.
- Always clearly specify your problem with the error message or screenshots of your code/results/error, or nobody could help you.
- Start early!!!







- ◆ 所有公告會公告於fb社團或寄信給各位。
- 請於報告前一天晚上23:59:59之前把報告上傳到
 <u>https://reurl.cc/4R5kDv</u>,統一使用教室電腦報告