HW1改進與調整

交大電機碩一 張詔揚

Espnet 資料目錄

- espnet/ # Python modules
- egs2/ # The complete recipe for each corpora
- [name]/
- asr1/ # ASR recipe
- downloads/
- resource_aishell/
- -speaker.info
- -lexicon.txt
- data_aishell #放訓練資料
- …

訓練資料準備

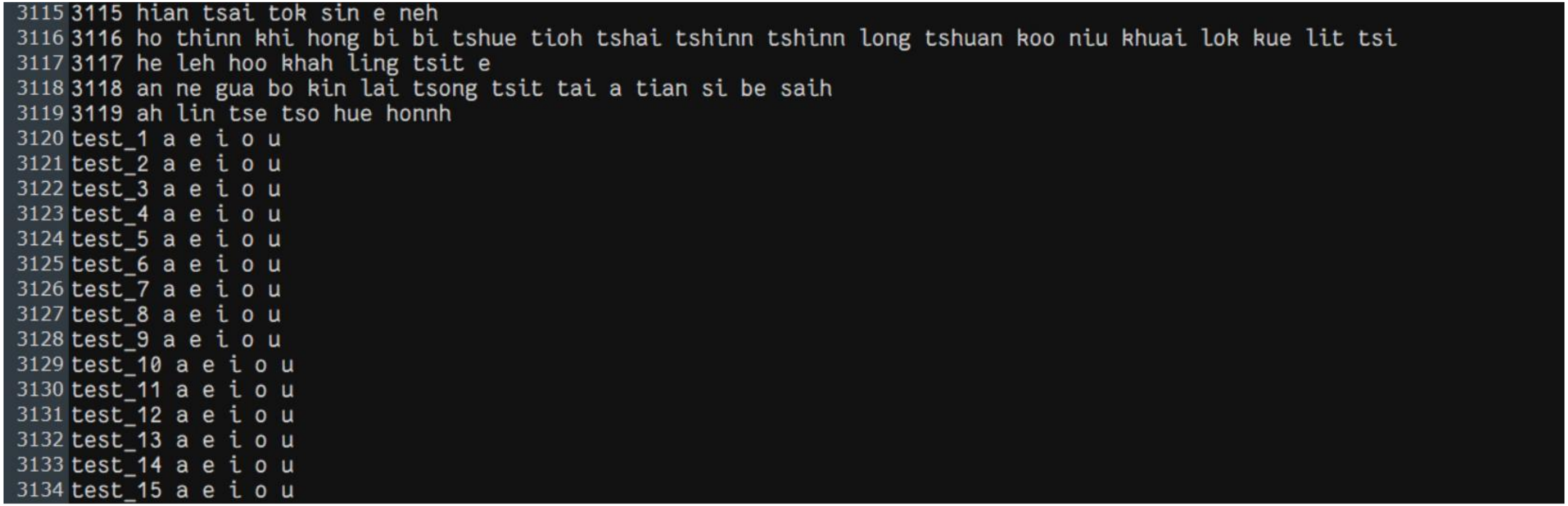
- data_aishell/
- wav/
- train/
- -global/ # 訓練音檔
- test/
- -global/ # Kaggles競賽用音檔
- -dev/
- -global/ # 測試音檔(不能是空的)
- transcript/
- aishell_transcript.txt #放與音檔對應的文字檔

怎麼產生我們要的文字檔

- downloads/
- resource_aishell/
- data_aishell
- wav/
- transcript/
- - aishell_transcript.txt

transcript.txt的擺放方式

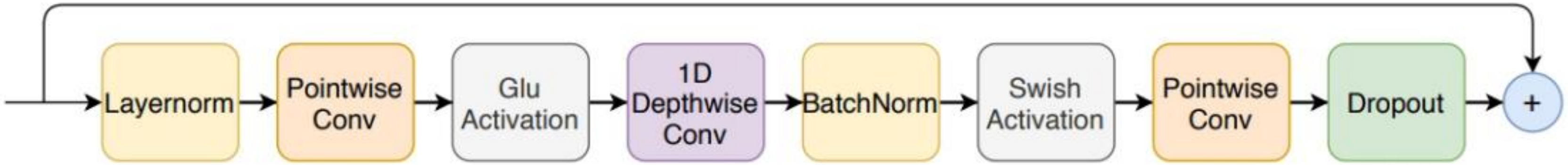
- 為了生成文字檔，必須把想要辨認的音檔文字改成 a e i o u之後加到transcript.txt的末尾



```
3115 hian tsai tok sin e neh
3116 ho thinn khi hong bi bi tshue tioh tshai tshinn tshinn long tshuan koo niu khuai lok kue lit tsi
3117 he leh hoo khah ling tsit e
3118 an ne gua bo kin lai tsong tsit tai a tian si be saih
3119 ah lin tse tso hue honnh
test_1 a e i o u
test_2 a e i o u
test_3 a e i o u
test_4 a e i o u
test_5 a e i o u
test_6 a e i o u
test_7 a e i o u
test_8 a e i o u
test_9 a e i o u
test_10 a e i o u
test_11 a e i o u
test_12 a e i o u
test_13 a e i o u
test_14 a e i o u
test_15 a e i o u
```

conformer架構

- 引入CNN 和 multi-head self attention的transformer架構
- multi-head self attention: 利用多個尺度的特徵來對長序列進行建模。這種機制可以更好地處理長序列，提高模型的性能和效率。
- separable convolution: 將卷積拆分成兩個部分，pointwise Conv & depthwise Conv ，可以大幅減少計算量



conformer架構圖

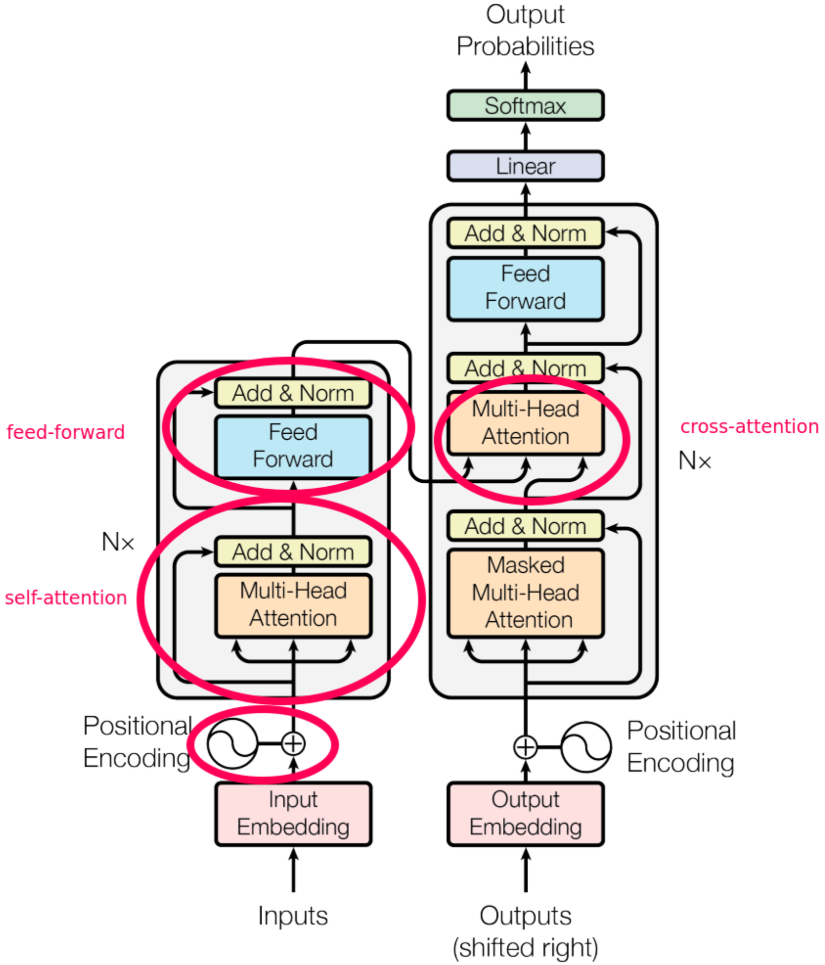
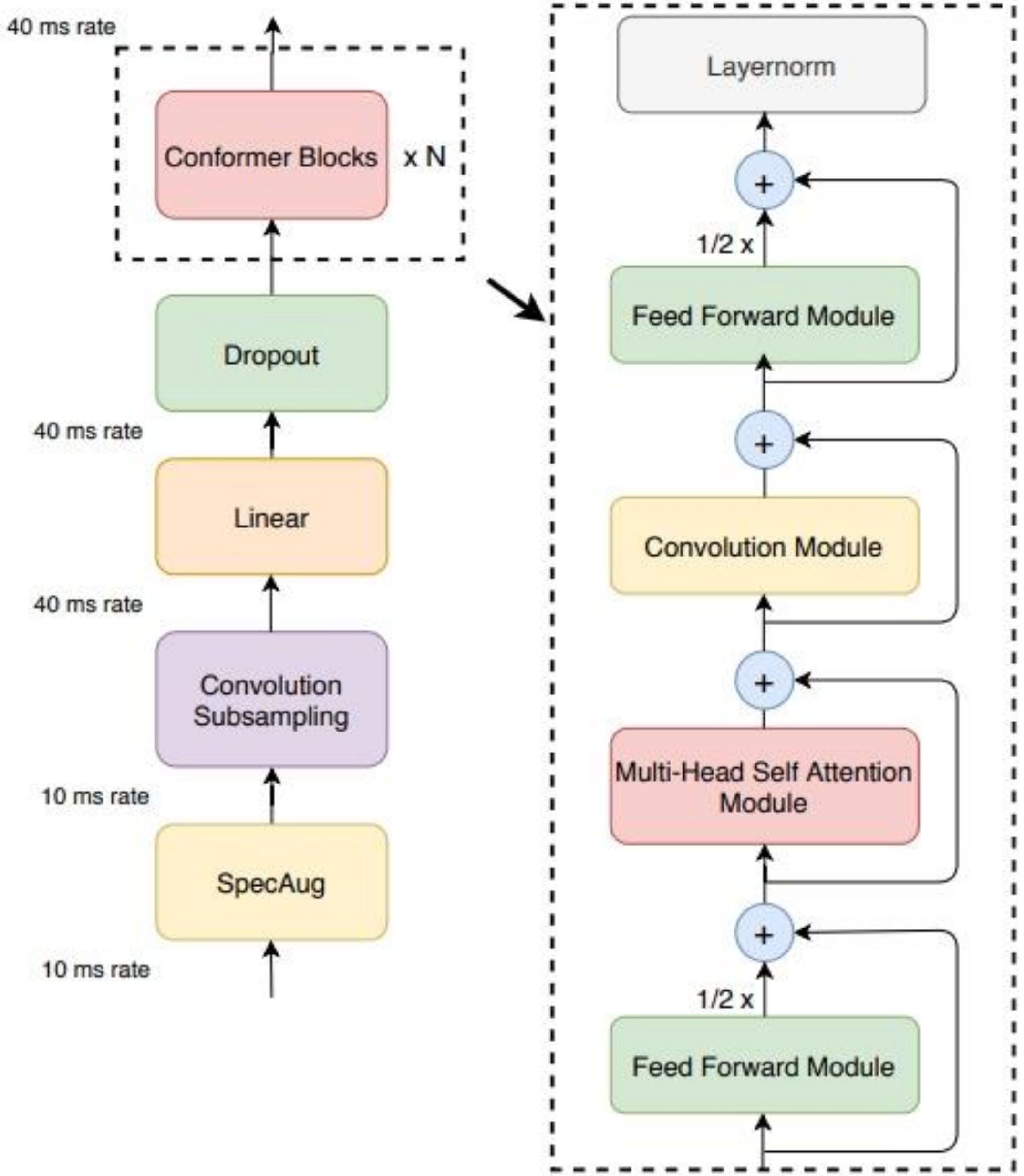
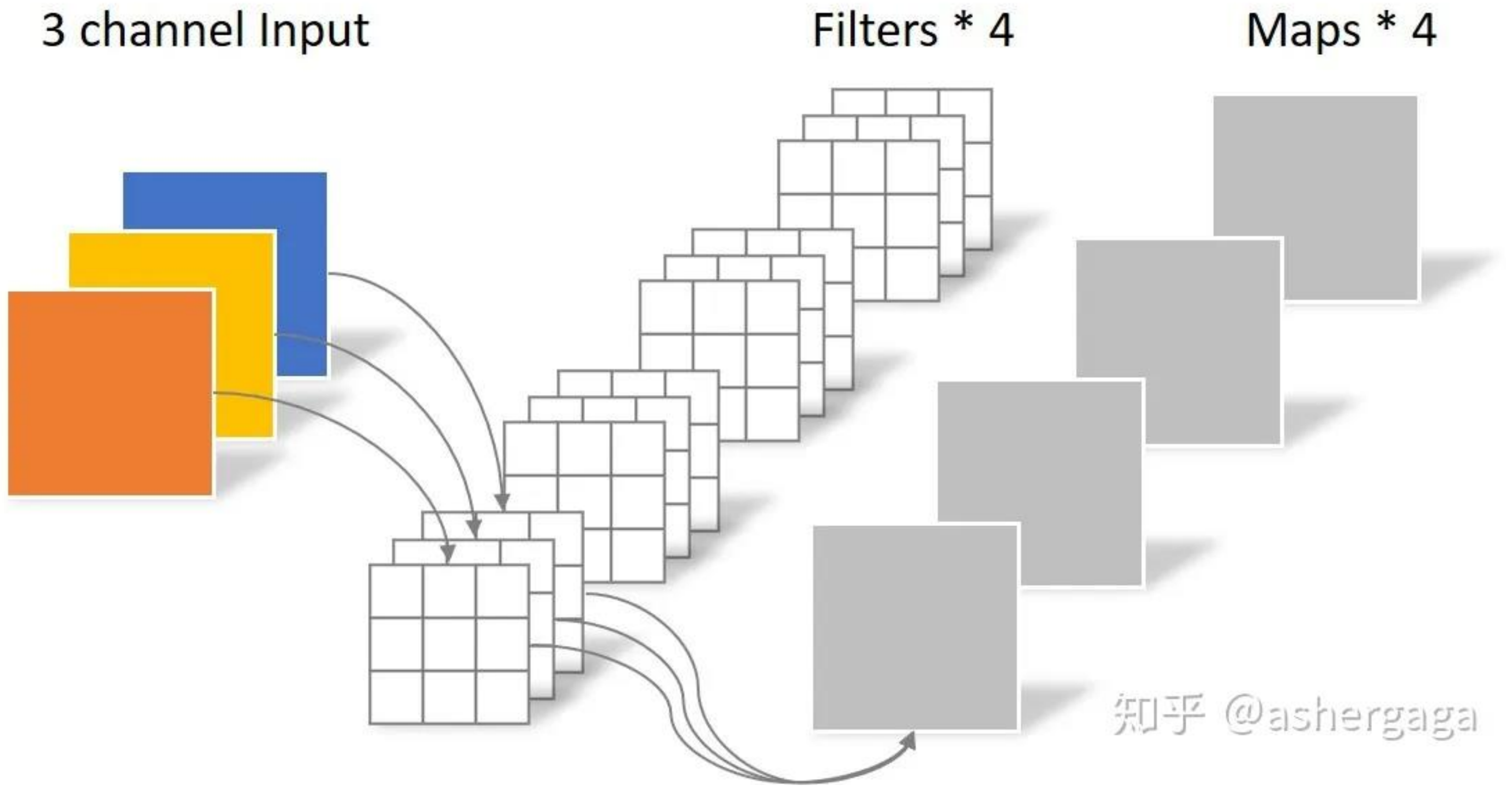


Figure 1: The Transformer - model architecture.



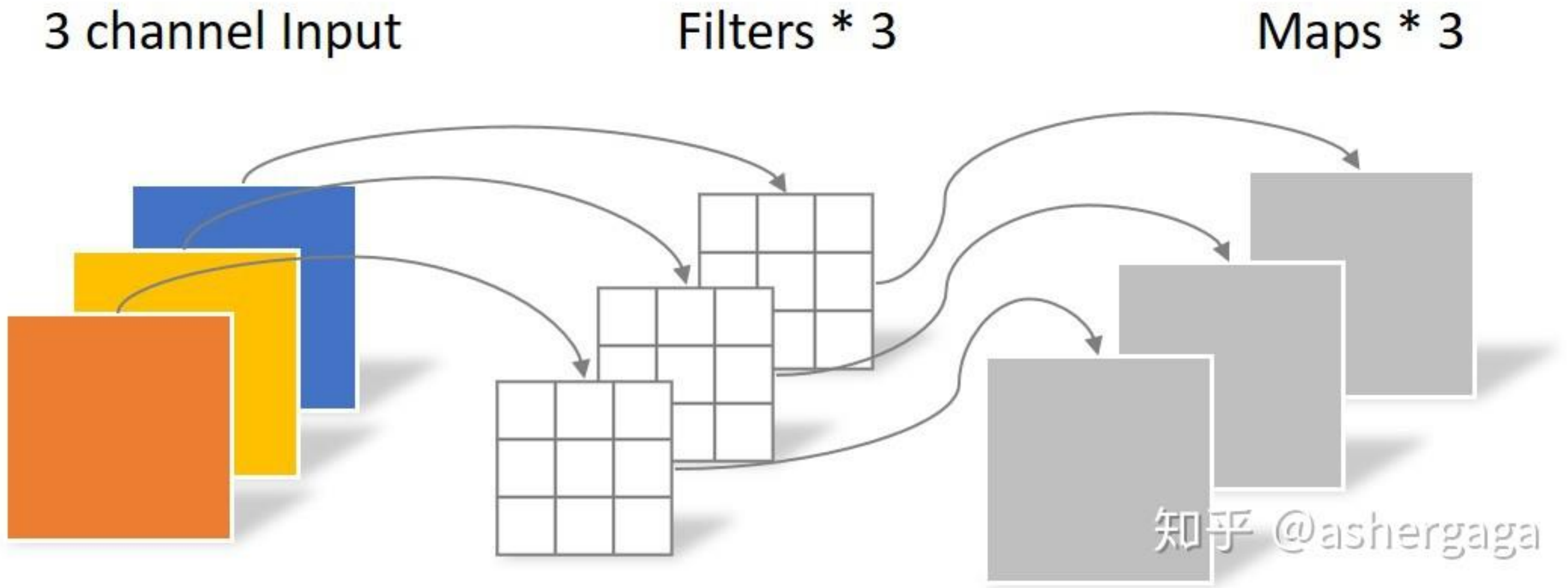
Depwise Conv & Pointwise Conv

- 參數量: N_std = 4 × 3 × 3 × 3 = 108



Depthwise Conv

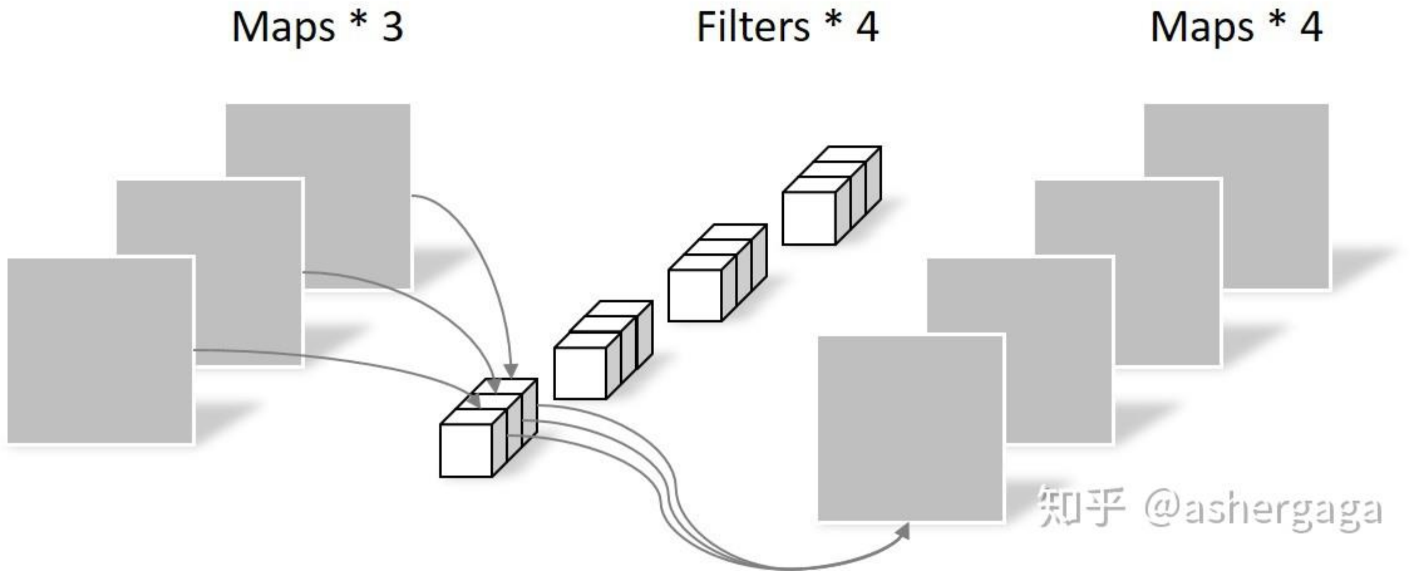
- 參數量: N_depthwise = 3 × 3 × 3 = 27



- depthwise 只改變feature map的大小，不改變通道數。

Pointwise Conv

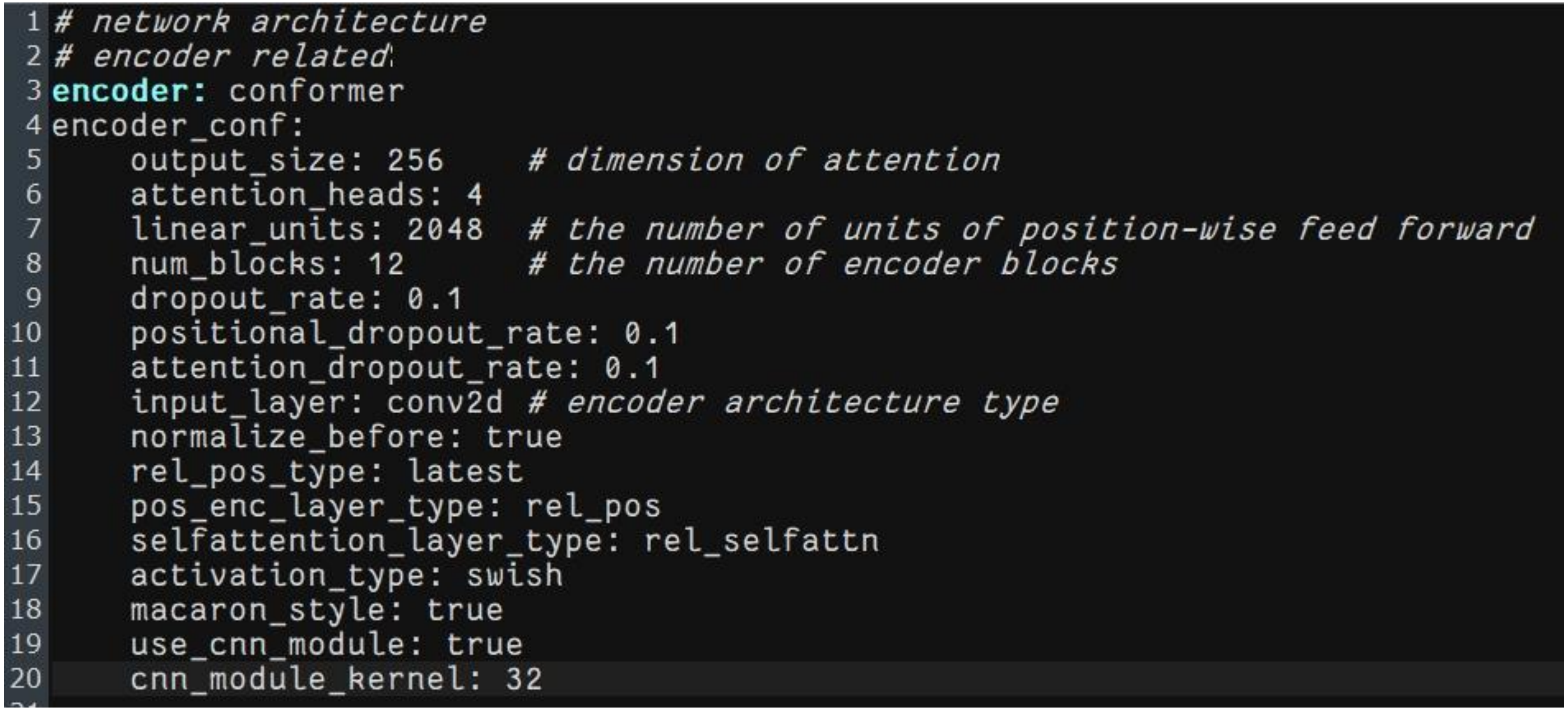
- 參數量: N_pointwise = 1 × 1 × 3 × 4 = 12



- pointwise指改變通道數，不改變feature map大小。
- N_separable = N_depthwise + N_pointwise = 39

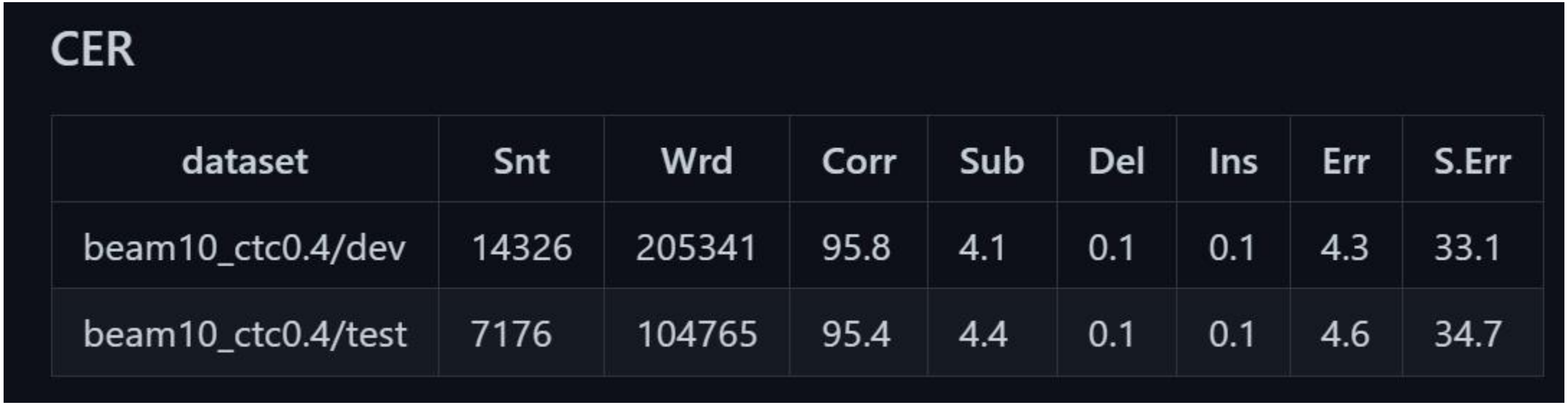
Espnet 配置文件

- conformer 模型的配置文件 train.yaml



```
# network architecture
# encoder related
encoder: conformer
encoder_conf:
    output_size: 256    # dimension of attention
    attention_heads: 4
    linear_units: 2048  # the number of units of position-wise feed forward
    num_blocks: 12      # the number of encoder blocks
    dropout_rate: 0.1
    positional_dropout_rate: 0.1
    attention_dropout_rate: 0.1
    input_layer: conv2d # encoder architecture type
    normalize_before: true
    rel_pos_type: latest
    pos_enc_layer_type: rel_pos
    selfattention_layer_type: rel_selfattn
    activation_type: swish
    macaron_style: true
    use_cnn_module: true
    cnn_module_kernel: 32
```

Conformer的表現



CER

dataset	Snt	Wrd	Corr	Sub	Del	Ins	Err	S.Err
beam10_ctc0.4/dev	14326	205341	95.8	4.1	0.1	0.1	4.3	33.1
beam10_ctc0.4/test	7176	104765	95.4	4.4	0.1	0.1	4.6	34.7

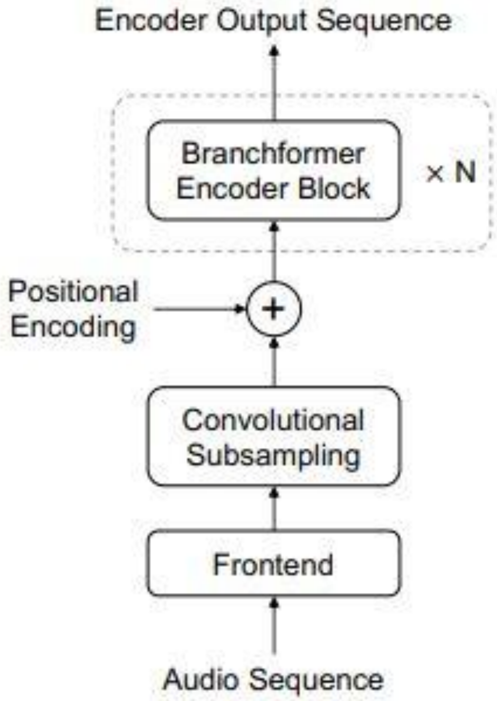
Branchformer 架構

- Branchformer**是將**global**和**local特徵**並行提取**，**相互之間是獨立的**，**兩種**特徵**最後可以進行結合**

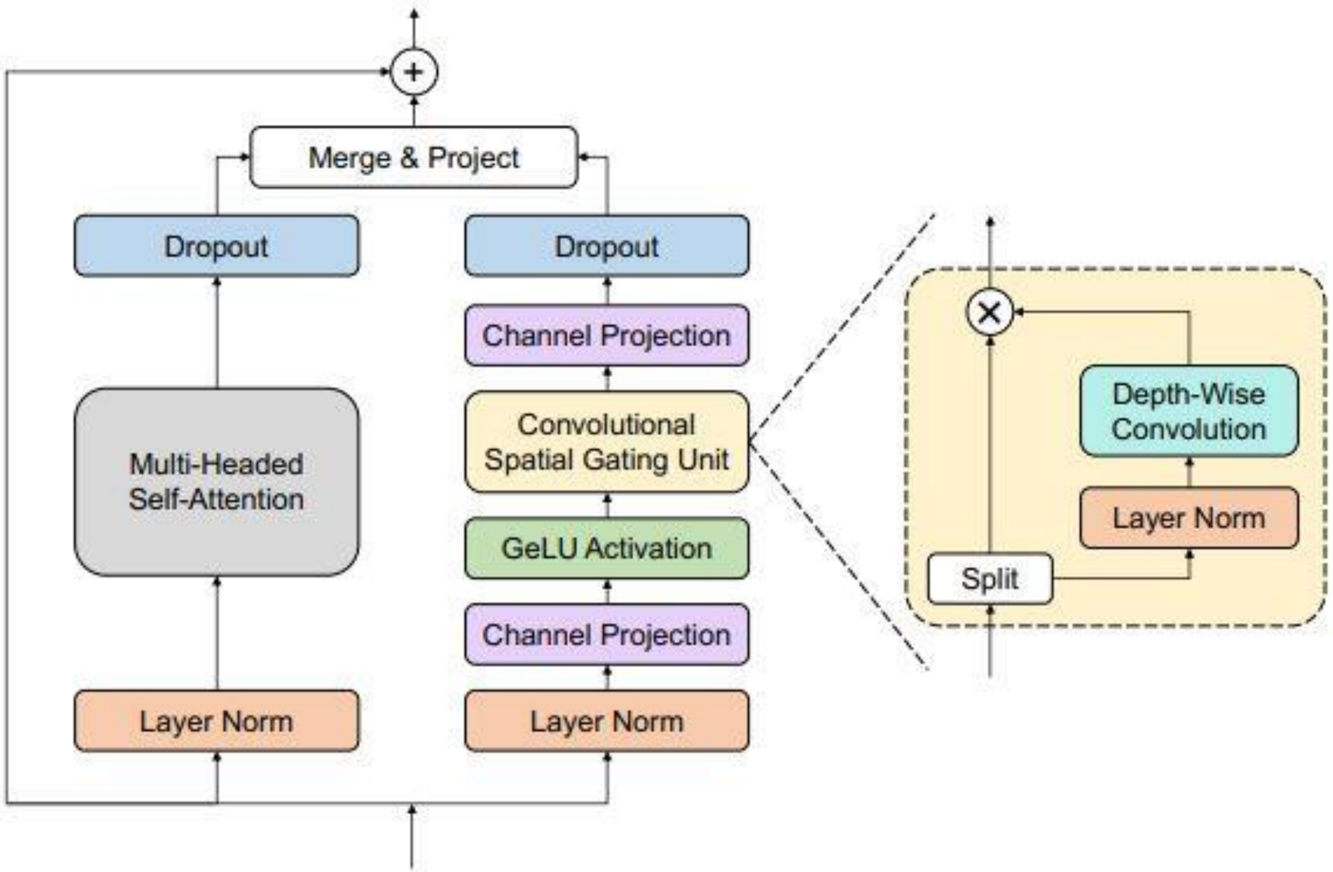
- branching mechanism: 將輸入序列分解成多個子序列，並且為每個子序列分配一個獨立的分支，進行獨立的自注意力(self-attention)計算。

- Branchformer 擅長處理長文本序列(因為將輸入序列分解成多個子序列)

Branchformer 架構圖

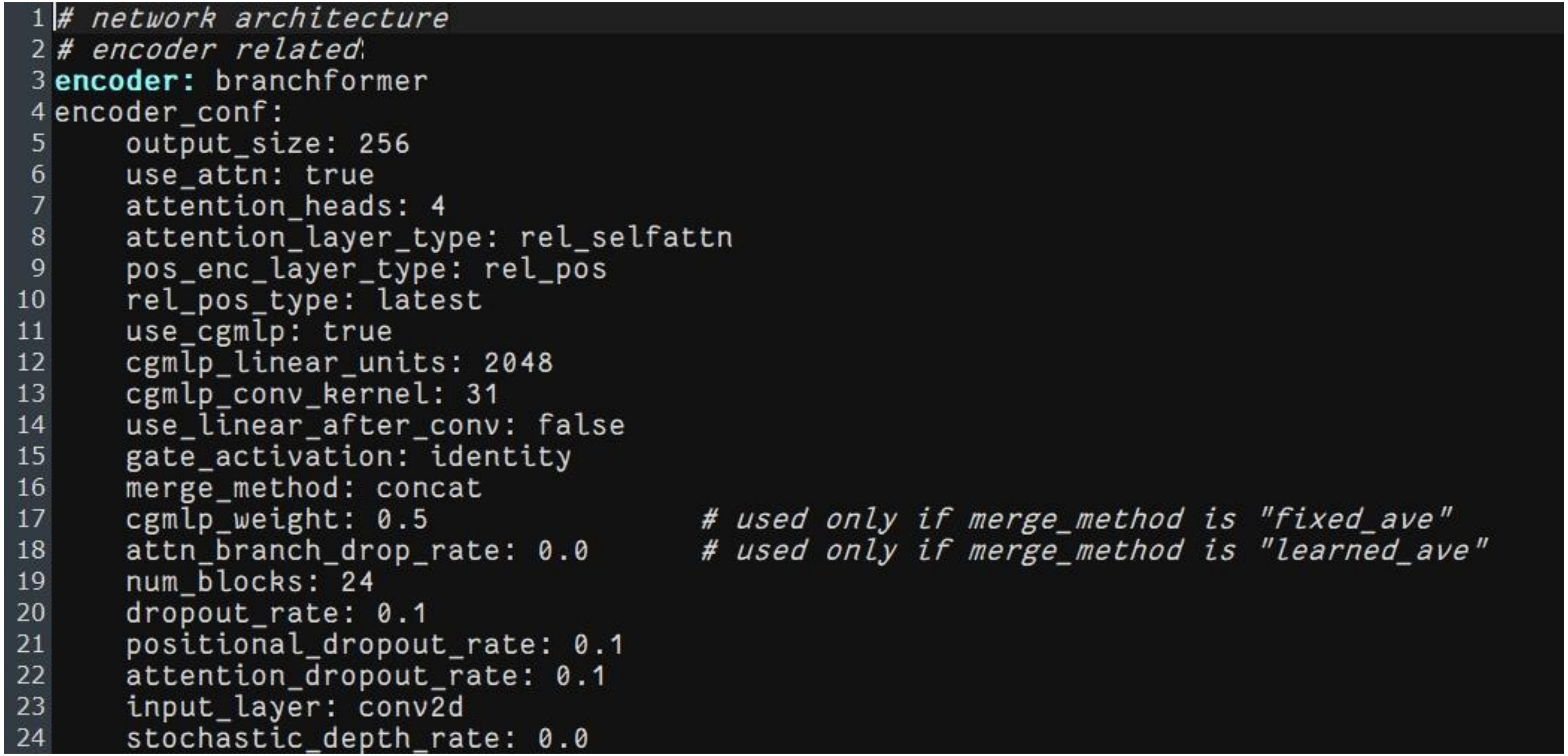


(a) Overall architecture of the encoder. A stack of identical Branchformer blocks are used to capture local and global dependencies.



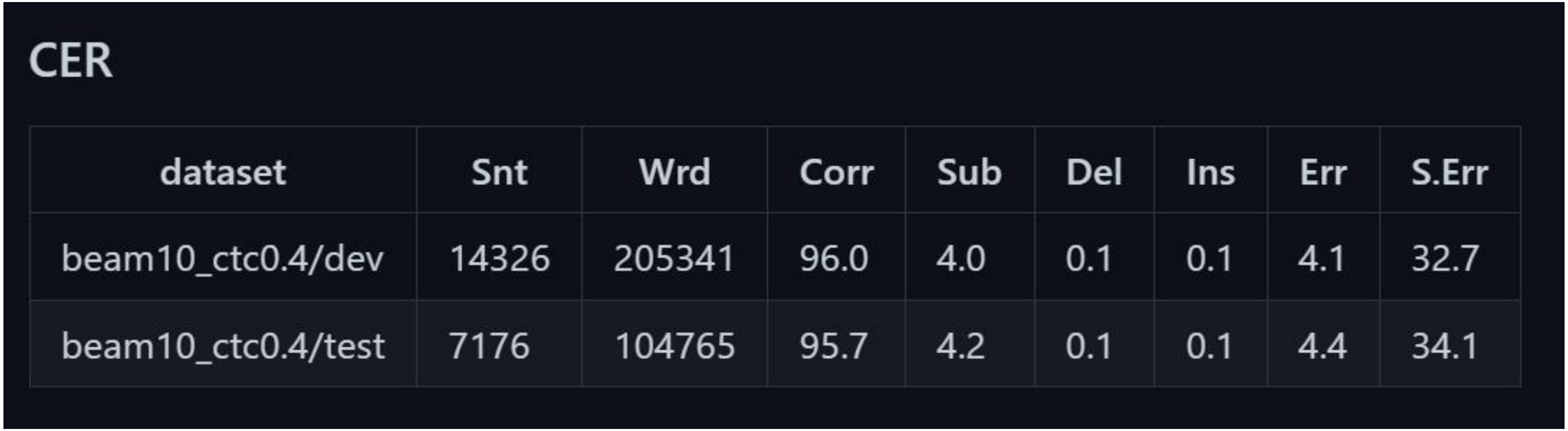
(b) Architecture of our Branchformer encoder block. It consists of two parallel branches. One branch employs attention to capture global context, while the other branch utilizes the MLP with convolutional gating to extract local context.

Espnet 配置文件



```yaml
# network architecture
# encoder related
encoder: branchformer
encoder_conf:
    output_size: 256
    use_attn: true
    attention_heads: 4
    attention_layer_type: rel_selfattn
    pos_enc_layer_type: rel_pos
    rel_pos_type: latest
    use_cgmlp: true
    cgmlp_linear_units: 2048
    cgmlp_conv_kernel: 31
    use_linear_after_conv: false
    gate_activation: identity
    merge_method: concat
    cgmlp_weight: 0.5               # used only if merge_method is "fixed_ave"
    attn_branch_drop_rate: 0.0      # used only if merge_method is "learned_ave"
    num_blocks: 24
    dropout_rate: 0.1
    positional_dropout_rate: 0.1
    attention_dropout_rate: 0.1
    input_layer: conv2d
    stochastic_depth_rate: 0.0
```

Branchformer的表現(官方網站)

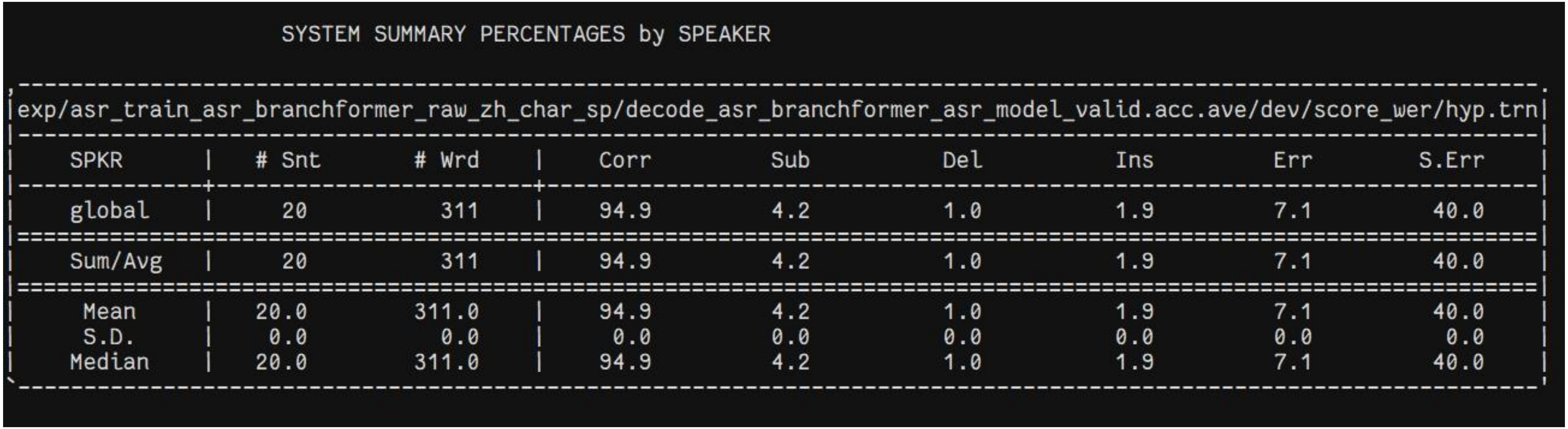


CER

dataset	Snt	Wrd	Corr	Sub	Del	Ins	Err	S.Err
beam10_ctc0.4/dev	14326	205341	96.0	4.0	0.1	0.1	4.1	32.7
beam10_ctc0.4/test	7176	104765	95.7	4.2	0.1	0.1	4.4	34.1

測試結果(自己的)

- 在解碼文件輸出目錄中，會生成一個result.txt 文件，該文件保存了最終在測試集和驗證集上的解碼結果，如下圖所示：

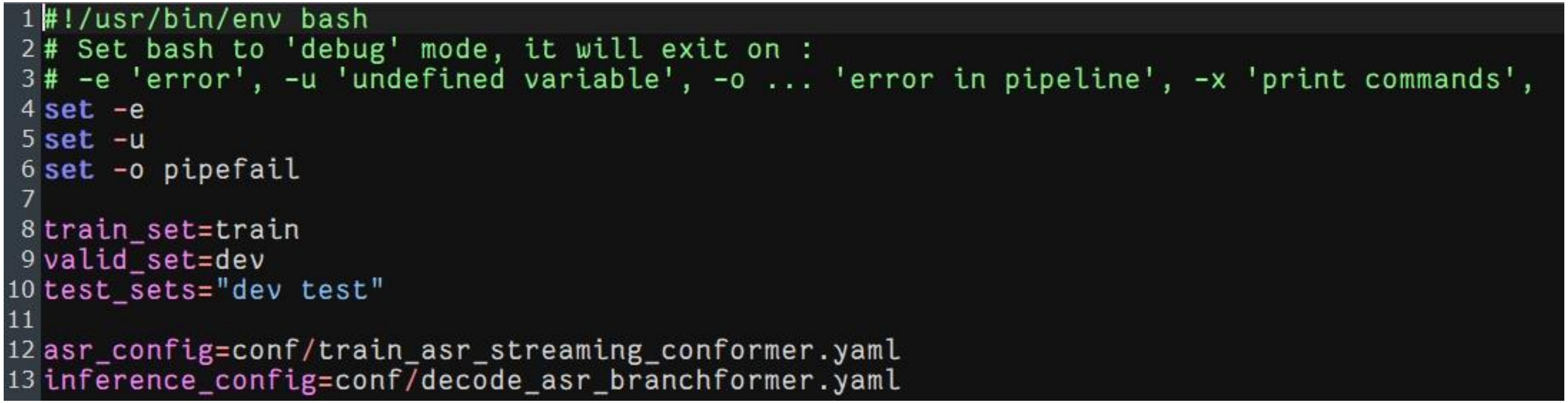


```
                     SYSTEM SUMMARY PERCENTAGES by SPEAKER

,-----------------------------------------------------------------------------------------------------------------.
|exp/asr_train_asr_branchformer_raw_zh_char_sp/decode_asr_branchformer_asr_model_valid.acc.ave/dev/score_wer/hyp.trn|
|-----------------------------------------------------------------------------------------------------------------|
|   SPKR        |  # Snt       # Wrd   |    Corr         Sub         Del         Ins         Err       S.Err     |
|---------------+----------------------+--------------------------------------------------------------------------|
|   global      |    20          311   |    94.9         4.2         1.0         1.9         7.1        40.0     |
|=================================================================================================================|
|   Sum/Avg     |    20          311   |    94.9         4.2         1.0         1.9         7.1        40.0     |
|=================================================================================================================|
|    Mean       |  20.0        311.0   |    94.9         4.2         1.0         1.9         7.1        40.0     |
|    S.D.       |   0.0          0.0   |     0.0         0.0         0.0         0.0         0.0         0.0     |
|   Median      |  20.0        311.0   |    94.9         4.2         1.0         1.9         7.1        40.0     |
`-----------------------------------------------------------------------------------------------------------------'
```

怎麼更換model

- 更改/[username]/espnet/egs2/aishell1/asr1/run.sh的文件



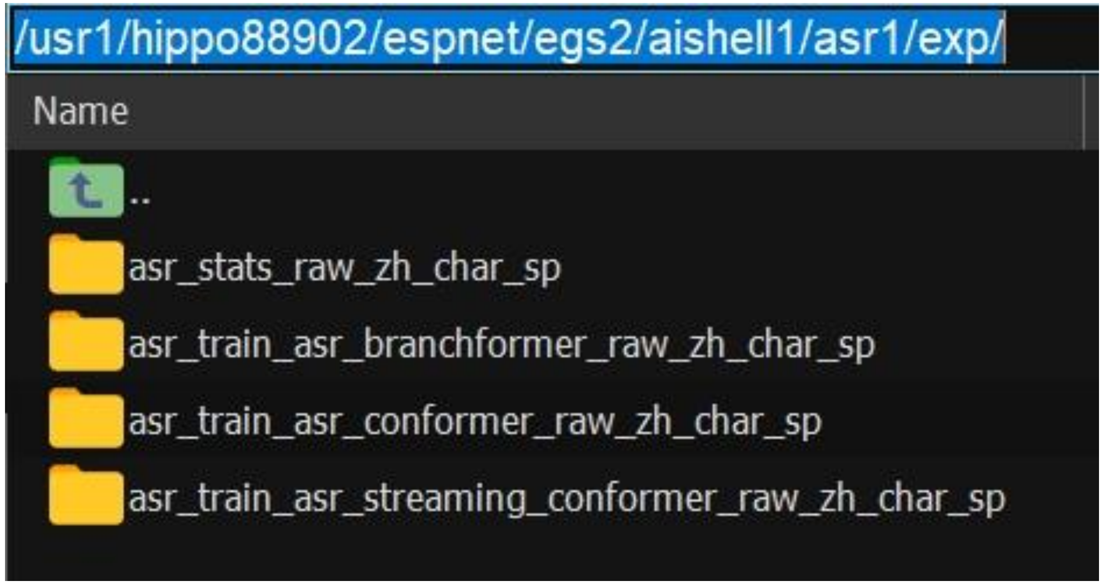
```
#!/usr/bin/env bash
# Set bash to 'debug' mode, it will exit on :
# -e 'error', -u 'undefined variable', -o ... 'error in pipeline', -x 'print commands',
set -e
set -u
set -o pipefail

train_set=train
valid_set=dev
test_sets="dev test"

asr_config=conf/train_asr_streaming_conformer.yaml
inference_config=conf/decode_asr_branchformer.yaml
```

不同model出來的結果放在哪

- /usr1/[username]/espnet/egs2/aishell1/asr1/exp/



```
/usr1/hippo88902/espnet/egs2/aishell1/asr1/exp/
Name
..
asr_stats_raw_zh_char_sp
asr_train_asr_branchformer_raw_zh_char_sp
asr_train_asr_conformer_raw_zh_char_sp
asr_train_asr_streaming_conformer_raw_zh_char_sp
```

- /decode_asr_branchformer_asr_model_valid.acc.ave/test/text 就是辨認後的文字檔