DLHLP HW3 Report

GitHub ID: PengWenChen
學號：R08942085 系級：電信丙 姓名：陳芃彣
學號：R08942086 系級：電信丙 姓名：趙達軒
學號：R07942091 系級：電信丙 姓名：許博閔
學號：R08945002 系級：生醫電資 姓名：陳玠妏

1. (5%)請記錄 evaluate.log 裡面的SiSNR 數值，和當時所用的 hyperparameter(這一題請3-1不用PIT, 3-2用PIT)

3-1

```
sample_rate=8000
segment=4  # seconds
cv_maxlen=6  # seconds
# Network config
N=128
L=40
B=128
H=256
P=3
X=7
R=1
norm_type=gLN
causal=0
mask_nonlinear='relu'
C=2
# Training config
use_cuda=1
id=0
epochs=100
half_lr=1
early_stop=1
max_norm=5
pit=0
# minibatch
shuffle=1
batch_size=30
num_workers=8
# optimizer
optimizer=adam
lr=1e-3
momentum=0
l2=0
```

```
Average SDR improvement: 17.60
Average SISNR improvement: 17.39
# Accounting: time=120 threads=1
```

3-2

```
sample_rate=8000
segment=4  # seconds
cv_maxlen=6  # seconds
# Network config
N=256 #128
L=40 #40 40
B=128 #128
H=256 #256
P=3 #3
X=8 #7 8
R=3 #1 3
norm_type=gLN
causal=0
mask_nonlinear='relu'
C=2
# Training config
use_cuda=1
id=0
epochs=120
half_lr=1
early_stop=1
max_norm=5
pit=1
# minibatch
shuffle=1
batch_size=30
num_workers=4
# optimizer
optimizer=adam
lr=1e-3 # 1e-3
momentum=0
l2=0
```

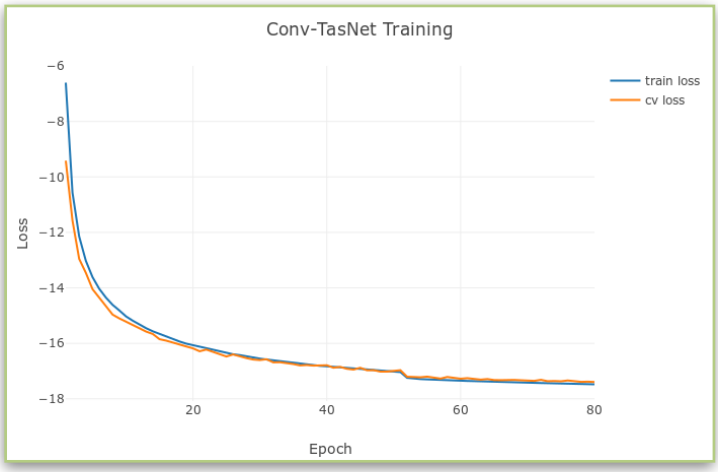
```
Average SDR improvement: 10.75
Average SISNR improvement: 10.20
# Accounting: time=59 threads=1
```

2. (5%)嘗試調整不同的hyperparameter，比較其差異，並試著分析結果(至少針對2種不同的hyperparameter進行實驗)

3-1

針對第一題，我們選用了與第一題的model做比較。首先，我們將X跟R調大，看看會發生甚麼事。

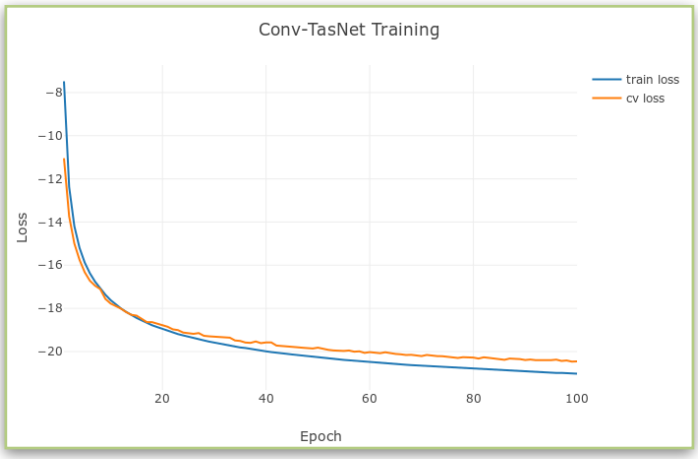
原本參數：




```
Average SDR improvement: 17.60
Average SISNR improvement: 17.39
# Accounting: time=120 threads=1
```

```
24 sample_rate=8000
25 segment=4  # seconds
26 cv_maxlen=6  # seconds
27 # Network config
28 N=128
29 L=40
30 B=128
31 H=256
32 P=3
33 X=7
34 R=1
35 norm_type=gLN
36 causal=0
37 mask_nonlinear='relu'
38 C=2
39 # Training config
40 use_cuda=1
41 id=0
42 epochs=100
43 half_lr=1
44 early_stop=1
45 max_norm=5
46 pit=0
47 # minibatch
48 shuffle=1
49 batch_size=30
50 num_workers=8
51 # optimizer
52 optimizer=adam
53 lr=1e-3
54 momentum=0
55 l2=0
```

調整過後參數：




```
264  Average SDR improvement: 20.96
265  Average SISNR improvement: 20.82
266  # Accounting: time=37 threads=1
```

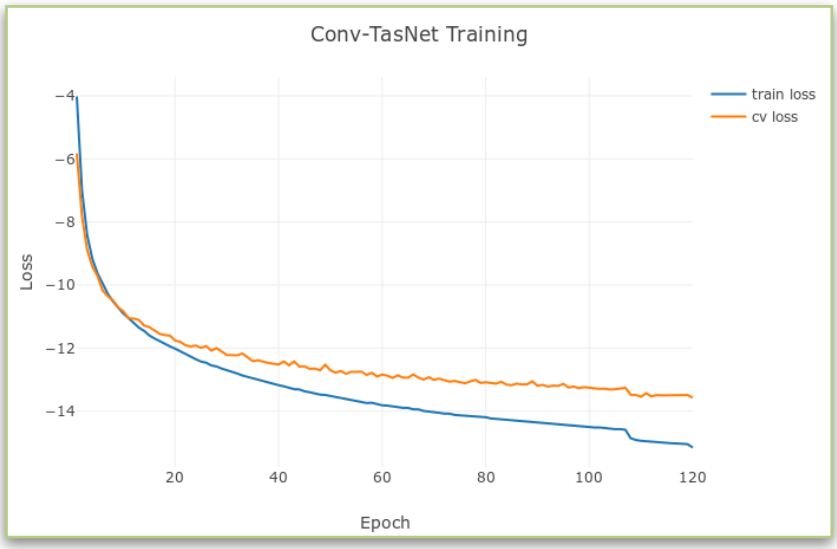
```
25  sample_rate=8000
26  segment=4  # seconds
27  cv_maxlen=6  # seconds
28  # Network config
29  N=128 #128
30  L=40 # 40
31  B=128 #128
32  H=256 #256
33  P=3 #3
34  X=8 #7 8
35  R=3 #1 3
36  norm_type=gLN
37  causal=0
38  mask_nonlinear='relu'
39  C=2
40  # Training config
41  use_cuda=1
42  id=0
43  epochs=100
44  half_lr=1
45  early_stop=1
46  max_norm=5
47  pit=0
48  # minibatch
49  shuffle=1
50  batch_size=24
51  num_workers=8
52  # optimizer
53  optimizer=adam
54  lr=1e-3 # 1e-3
55  momentum=0
56  l2=0
```

我們發現把X跟R調大，結果會變好。其中，R代表 Tasnet 中有幾組 convolution block，然後X代表每組有幾個convolution block。我們發現越多convolution block可以截取越多資訊，從而讓分類效果更好。

3-2

我們將3-2的參數做了一些調整。以下是兩種不同參數的比較：

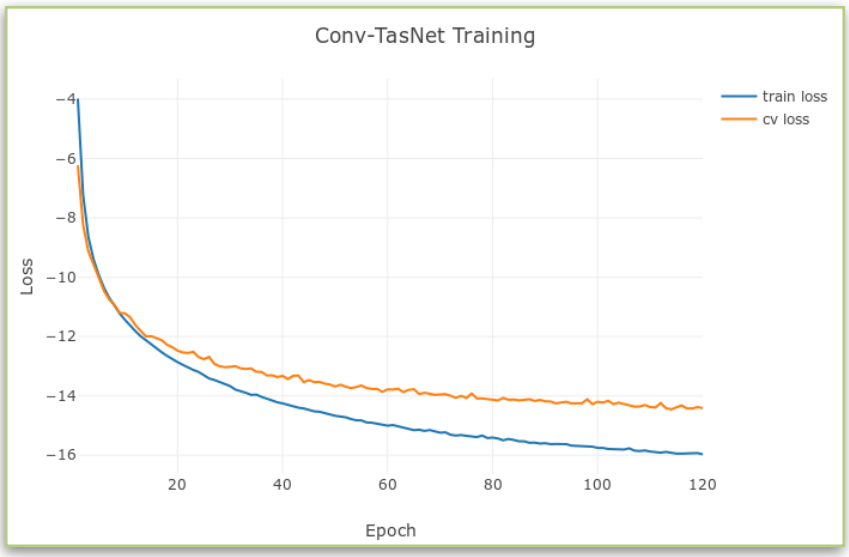
以下是第一種參的結果：




```
Average SDR improvement: 10.32
Average SISNR improvement: 9.74
# Accounting: time=24 threads=1
```

```
sample_rate=8000
segment=4  # seconds
cv_maxlen=6  # seconds
# Network config
N=256 #128
L=40 #40 40
B=128 #128
H=256 #256
P=3 #3
X=7 #7 8
R=2 #1 3
norm_type=gLN
causal=0
mask_nonlinear='relu'
C=2
# Training config
use_cuda=1
id=0
epochs=120
half_lr=1
early_stop=1
max_norm=5
pit=1
# minibatch
shuffle=1
batch_size=30
num_workers=4
```

後來，我們將X跟R調大，結果如下：




```
Average SDR improvement: 10.75
Average SISNR improvement: 10.20
# Accounting: time=59 threads=1
```

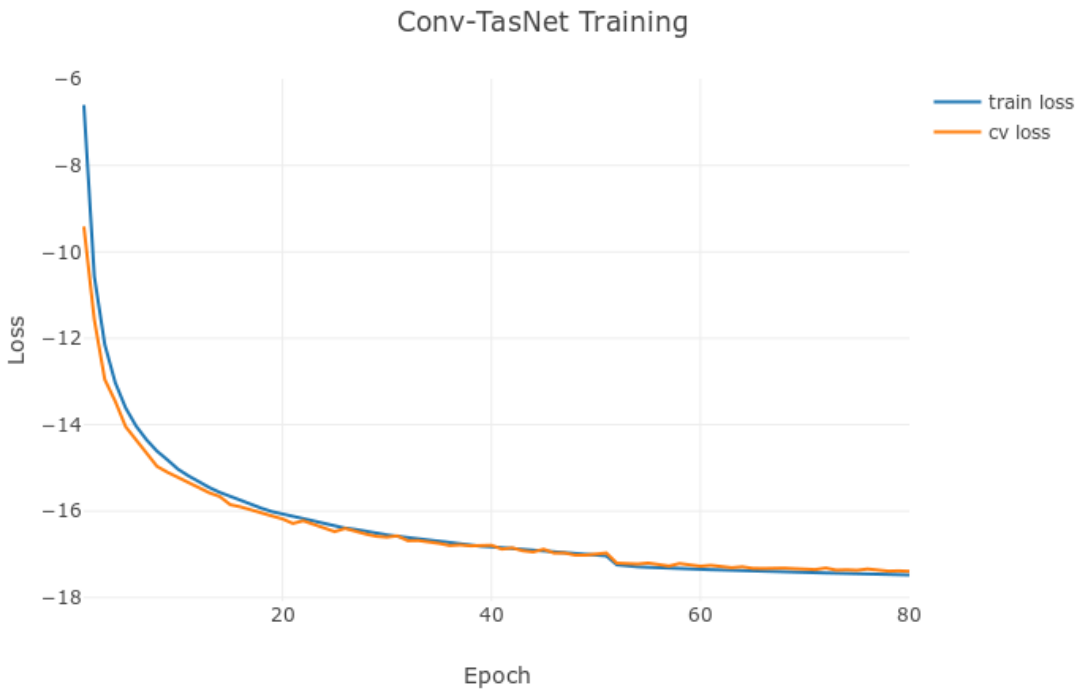
```
sample_rate=8000
segment=4  # seconds
cv_maxlen=6  # seconds
# Network config
N=256 #128
L=40 #40 40
B=128 #128
H=256 #256
P=3 #3
X=8 #7 8
R=3 #1 3
norm_type=gLN
causal=0
mask_nonlinear='relu'
C=2
# Training config
use_cuda=1
id=0
epochs=120
half_lr=1
early_stop=1
max_norm=5
pit=1
# minibatch
shuffle=1
batch_size=30
num_workers=4
# optimizer
optimizer=adam
lr=1e-3 # 1e-3
momentum=0
l2=0
```

我們發現把X跟R調大，結果會變好，而且就過Baseline了。其中，R代表Tasnet中有幾組convolution block，然後X代表每組有幾個convolution block。我們發現越多convolution block可以截取越多資訊，從而讓分類效果更好。

3. (3%)3-1, 3-2請分別試看看有無PIT的差異並記錄結果(loss learning curve, Si-SNR)

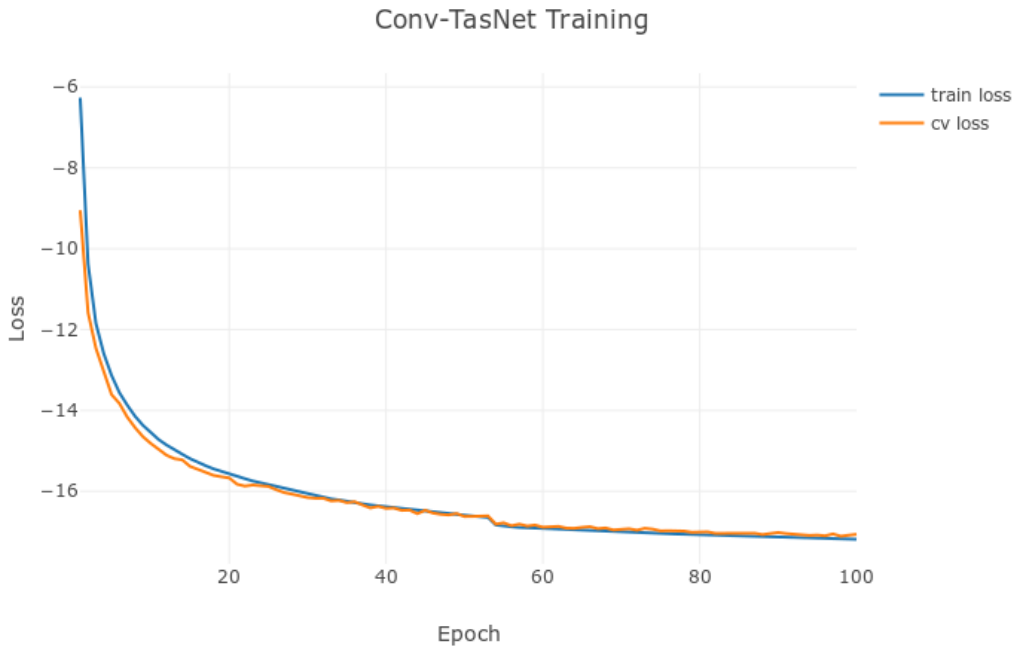
3-1

PIT = 0




```
Average SDR improvement: 17.60
Average SISNR improvement: 17.39
# Accounting: time=120 threads=1
```

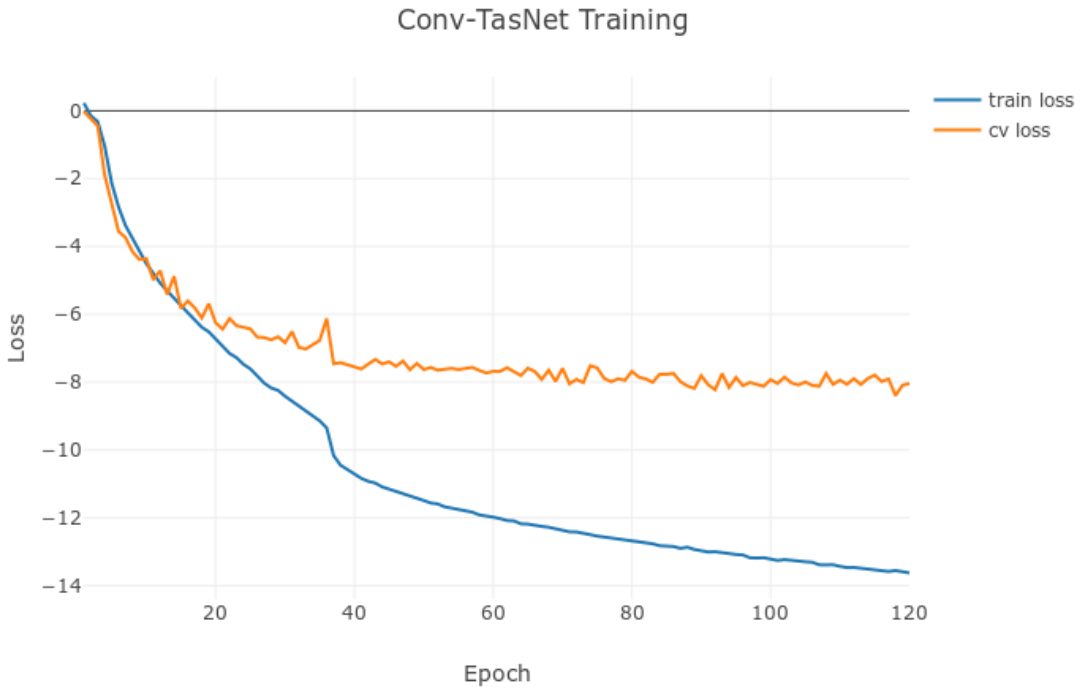
PIT = 1




```
Average SISNR improvement: 17.25
# Accounting: time=35 threads=1
```

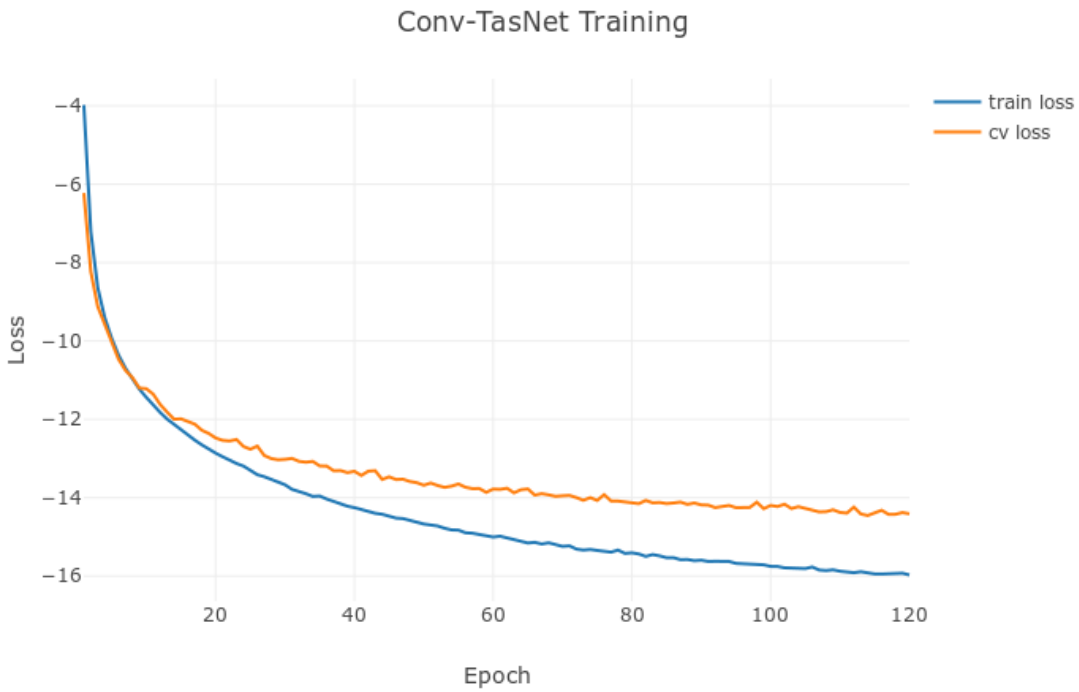
3-2

PIT=0




```
Average SISNR improvement: 3.05
# Accounting: time=31 threads=1
```

PIT=1




```
Average SISNR improvement: 10.20
# Accounting: time=59 threads=1
```

4. (2%)思考一下為何有無PIT會影響3-1, 3-2的結果並寫下你的看法

3-1 有沒有加 PIT 結果都差不多，但是3-2有沒有加PIT卻差很多。我們分析，原因是3-1只有兩個人而已，所以只有兩種排列，就是[0,1]跟[1,0]。所以就算是沒有加PIT，3-1原本訓練的方式也是這樣的。但是3-2因為分成多個音檔，所以利用PIT能夠計算出誤差最小的排列組合，所以比起沒有PIT，他在分開音檔時能夠大幅提升效果。