

# Overlapping with language modelling

---

Code to reproduce the results from "Alleviating Sequence Information Loss with Data Overlapping and PrimeBatch Sizes".

The taxonomy in the code may differ a bit from the paper, especially regarding the type of experiments. Here is the corresponding terms:

In the code	In the paper
No order	Extreme TOI
Local order	Inter-batch TOI
Standard order	Standard TOI
Total order (P)	Alleviated TOI (P)

Hold experiments on 4 models using the overlapping:

- **simple**, a very basic lstm for language modelling
- **awd**, [AWD](#) ASGD Weight-Dropped LSTM
- **mos**, [MOS](#) Mixture of Softmaxes
- **emotion**, a very basic LSTM for emotion detection on voice

To specify which model to run, use `--main-model {simple-lstm | awd-lstm | mos-lstm | emotions-simple-lstm}` argument. There are additional common parameters, as well as specific parameters for each model. Those can be found in `main_run.py`.

## Set-up

Download the data (PTB, WT2, WT103):

```
chmod +x get_data.sh
./get_data.sh
```

For emotions, add in `data/IEMOCAP/` the `all_features_cv` files.

We use python `3.6` with Pytorch `0.4.1`. To create a new python environment and install dependencies, run:

```
python3.6 -m virtualenv venv
source venv/bin/activate
pip3 install -r requirements.txt
```

## About the files

`main_run.py` is the main entry point that parses arguments, does the global initialization and runs the corresponding model and task.

`awd/`, `emotions/`, `mos/` and `simple/` are the different models directories. `common/` holds the common initialization and utilities, such as the different data iterators, which are in the `DataSelector` class in `common/excavator.py`.

The `main_run.py` file, after performing the common initializations, imports the `main.py` file corresponding to the chosen model.

## Commands to reproduce the experiments

---

### AWD PTB

#### Extreme TOI:

```
python3 main_run.py --main-model awd-lstm --batch-size 20 --data data/penn
--dropouti 0.4 --dropout 0.25 --seed 141 --seed-shuffle 141 --epoch 1000
--shuffle-full-seq
```

#### Inter-batch TOI:

```
python3 main_run.py --main-model awd-lstm --batch-size 20 --data data/penn
--dropouti 0.4 --dropout 0.25 --seed 141 --seed-shuffle 141 --epoch 1000
--shuffle-row-seq
```

#### Standard TOI:

```
python3 main_run.py --main-model awd-lstm --batch-size 20 --data data/penn
--dropouti 0.4 --dropout 0.25 --seed 141 --epoch 1000
```

#### Alleviated TOI {2,5,7,10}:

```
overlaps=(2 5 7 10)
epochs=1000
for k in "${overlaps[@]}"
do
:
python3 main_run.py --main-model awd-lstm --batch-size 20 --data
data/penn --dropouti 0.4 --dropout 0.25 --seed 141 --epoch
"$((($epochs/$k))" --init-seq "overlapCN_${k}"
sleep 10
done
```

## AWD WT2

### Extreme TOI

```
python3 main_run.py --main-model awd-lstm --epochs 750 --data
/data/noemien.kocher/datasets/wikitext-2 --dropouth 0.2 --seed 1882 --
batch-size 80 --shuffle-full-seq
```

### Inter-batch TOI

```
python main_run.py --main-model awd-lstm --epochs 750 --data
/data/noemien.kocher/datasets/wikitext-2 --dropouth 0.2 --seed 1882 --
batch-size 80 --shuffle-row-seq
```

### Standard TOI

```
python3 main_run.py --main-model awd-lstm --epochs 750 --data
/data/noemien.kocher/datasets/wikitext-2 --dropouth 0.2 --seed 1882 --
batch-size 80
```

### Alleviated TOI {2,5,7,10}

```
overlaps=(2 5 7 10)
epochs=750
for k in "${overlaps[@]}"
do
:
python3 main_run.py --main-model awd-lstm --data
/data/noemien.kocher/datasets/wikitext-2 --dropouth 0.2 --seed 1882 --
batch-size 80 --epochs "$(($epochs/$k))" --init-seq "overlapCN_${k}"
sleep 10
done
```

## AWD WT103

### Extreme TOI

```
python3 -u main_run.py --main-model awd-lstm --epochs 14 --nlayers 4 --
emsize 400 --nhid 2500 --alpha 0 --beta 0 --dropoute 0 --dropouth 0.1 --
dropouti 0.1 --dropout 0.1 --wdrop 0 --wdecay 0 --bptt 140 --batch-size 60
--optimizer adam --lr 1e-3 --data /data/noemien.kocher/datasets/wikitext-
103 --when 12 --model QRNN --shuffle-full-seq
```

## Inter-batch TOI

```
python3 -u main_run.py --main-model awd-lstm --epochs 14 --nlayers 4 --
emsize 400 --nhid 2500 --alpha 0 --beta 0 --dropoute 0 --dropouth 0.1 --
dropouti 0.1 --dropout 0.1 --wdrop 0 --wdecay 0 --bptt 140 --batch-size 60
--optimizer adam --lr 1e-3 --data /data/noemien.kocher/datasets/wikitext-
103 --when 12 --model QRNN --shuffle-row-seq
```

## Standard TOI

```
python3 -u main_run.py --main-model awd-lstm --epochs 14 --nlayers 4 --
emsize 400 --nhid 2500 --alpha 0 --beta 0 --dropoute 0 --dropouth 0.1 --
dropouti 0.1 --dropout 0.1 --wdrop 0 --wdecay 0 --bptt 140 --batch-size 60
--optimizer adam --lr 1e-3 --data /data/noemien.kocher/datasets/wikitext-
103 --when 12 --model QRNN
```

## Alleviated TOI {2,5,7,10}

```
# base num epochs is 14
overlaps=(2 5 7 10)
when_steps=147456
max_steps=172032
for i in "${!overlaps[@]}"
do
    :
    python3 -u main_run.py --main-model awd-lstm --epochs 14 --nlayers
4 --emsize 400 --nhid 2500 --alpha 0 --beta 0 --dropoute 0 --dropouth 0.1
--dropouti 0.1 --dropout 0.1 --wdrop 0 --wdecay 0 --bptt 140 --batch-size
60 --optimizer adam --lr 1e-3 --data
/data/noemien.kocher/datasets/wikitext-103 --when-steps "$when_steps" --
model QRNN --init-seq "overlapCN_${overlaps[$i]}" --log-dir
/data/noemien.kocher/logs/ --max-steps "$max_steps"
    sleep 10
done
```

## Simple PTB

### Extreme TOI:

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 20
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1500 --lr-decay 1 --shuffle-
full-seq
```

### Inter-batch TOI:

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 20
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1500 --lr-decay 1 --shuffle-
row-seq
```

### Standard TOI:

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 20
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1500 --lr-decay 1
```

### Alleviated TOI {2,5,7,10}:

```
overlaps=(2 5 7 10)
epochs=100
for k in "${overlaps[@]}"
do
    :
    python3 main_run.py --main-model simple-lstm --epochs
"$((($epochs/$k))" --batch-size 20 --dropout 0.15 --nlayers 2 --bptt 70 --
nhid 1500 --lr-decay 1 --init-seq "overlapCN_{$k}"
    sleep 10
done
```

## Simple WT2

### Extreme TOI

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 80
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1150 --lr-decay 1 --data
/data/noemien.kocher/datasets/wikitext-2 --shuffle-full-seq
```

### Inter-batch TOI

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 80
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1150 --lr-decay 1 --data
/data/noemien.kocher/datasets/wikitext-2 --shuffle-row-seq
```

### Standard TOI

```
python3 main_run.py --main-model simple-lstm --epochs 100 --batch-size 80
--dropout 0.15 --nlayers 2 --bptt 70 --nhid 1150 --lr-decay 1 --data
/data/noemien.kocher/datasets/wikitext-2
```

## Alleviated TOI {2,5,7,10}

```
overlaps=(2 5 7 10)
epochs=100
for k in "${overlaps[@]}"
do
    :
    python3 main_run.py --main-model simple-lstm --epochs
"$((($epochs/$k))" --batch-size 80 --dropout 0.15 --nlayers 2 --bptt 70 --
nhid 1150 --lr-decay 1 --data /data/noemien.kocher/datasets/wikitext-2 --
init-seq "overlapCN_{$k}"
    sleep 10
done
```

## MOS PTB

### Standard TOI:

```
python main.py --main-model mos-lstm --data data/penn --dropouti 0.4 --
dropoutl 0.29 --dropouth 0.225 --seed 28 --batch_size 12 --lr 20.0 --epoch
1000 --nhid 960 --nhidlast 620 --emsize 280 --n_experts 15
```

### Alleviated TOI {2,5,7,10}:

```
python main.py --main-model mos-lstm --data data/penn --dropouti 0.4 --
dropoutl 0.29 --dropouth 0.225 --seed 28 --batch_size 12 --lr 20.0 --epoch
1000 --nhid 960 --nhidlast 620 --emsize 280 --n_experts 15 --init-seq
overlapCN_7
```

## Acknowledgements

Code is heavily borrowed from the following sources:

- simple-lstm ([simple/](https://github.com/deeplearningathome/pytorch-language-model)): <https://github.com/deeplearningathome/pytorch-language-model>
- awd-lstm ([awd/](https://github.com/salesforce/awd-lstm-lm)): <https://github.com/salesforce/awd-lstm-lm>
- mos-lstm: ([mos/](https://github.com/zihangdai/mos)) <https://github.com/zihangdai/mos>