

フロー

/home/<id>上で以下を行う

minicondaの導入

minicondaのinsall

```
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh
```

仮想環境の準備

仮想環境の作成 `-n` に環境名を設定する

```
conda create -n <envname> python=3.8.10
```

作成した仮想環境を確認

```
conda info -e
```

仮想環境の有効化

```
conda activate <envname>
```

毎bash 起動ごとに仮想環境を有効にするには ~/.bashrcに

```
conda activate <envname>
```

を付け足す

Fairseqのinstall

```
git clone https://github.com/pytorch/fairseq
cd fairseq
pip install --editable ./
```

残り必要なライブラリのinstall

```
pip install sentencepiece
pip install tensorboard
```

以降は /data/group1/<自分のID>のところで作業する

学習用の諸々を持ってくる

```
git clone https://github.com/horiso0921/finetuning-nttdialoguemodel.git
```

ドロップボックス <https://www.dropbox.com/s/k3ugxmr7nw6t86l/japanese-dialog-transformer-1.6B.pt?dl=0> から モデルをダウンロードしてきて

./finetuning-nttdialoguemodel/model/base におく

Jobスクリプト

以下は /data/group1/自分のIDに Job ディレクトリを作り、そこでこのスクリプトを置き実行させることが望ましい

```
#!/bin/bash -x
#PJM -L rscgrp=cx-small
#PJM -L node=1
#PJM -L elapse=12:00:00
#PJM -j
#PJM -S
#PJM -o log/fine.log

module load cuda/10.2.89_440.33.01 openmpi_cuda/4.0.4 nccl/2.7.3
eval "$(/miniconda3/bin/conda shell.bash hook)"

conda activate <envname>
nvidia-smi
ID=<自分のID>
ARRAY_emp_per=(`ls /data/group1/${ID}/finetuning-nttdialoguemodel/data/RawData`)
WARMUP_STEPS=(500)
LR=(1e-04)
BATCH_SIZE=(16)
PARAMS=()
ARRAY=()
for warmup in ${WARMUP_STEPS[@]}
do
    for lr in ${LR[@]}
    do
        for batch in ${BATCH_SIZE[@]}
        do
            param=${warmup},${lr},${batch}
            PARAMS+=("$param")
        done
    done
done

for model in ${ARRAY_emp_per[@]}
do
    for param in ${PARAMS[@]}
    do
        MODEL_NAME=${model},BASE,${param}
        ARRAY+=("$MODEL_NAME")
    done
done

MODEL_NAME=${ARRAY[$PJM_BULKNUM]}
IFS=' '
read -a arrstr <<<"$MODEL_NAME"
modelname=${arrstr[0]}
basemodel=${arrstr[1]}
warmup=${arrstr[2]}
lr=${arrstr[3]}
batch=${arrstr[4]}
dir=${modelname}_${basemodel}_${warmup}_${batch}_${lr}
mkdir -p ${dir}
bash /data/group1/${ID}/finetuning-nttdialoguemodel/src/train_fine_tuning.sh $modelname $basemodel $warmup $batch $lr > ${dir}/train_`date`
```

必ず自分のIDのところを書き換えること

Job実行は `pjsub <上のファイル名>` ができる

テスト時は

1. `pjsub --sparam wait-time=60 --interact -L rscgrp=cx-interactive` を実行してGPUを使用可能にして
2. finetuning-nttdialoguemodel/data/RawData で train等があるところに Testデータを置いて
3. _prepro_spm_test.py でトークナイズして
4. preprocess_test.sh <dataname> (srcにある) で バイナリ化して

5. `eval_test.sh`で結果を得る（結果は標準出力されます（エラー出力かも？））