# wav2vec



## 一、fairseq安装

Plain Text | 复制代码

```
Traceback (most recent call last):
  File "/home/zhuozhu.zz/tops/fairseq/bin/fairseq-hydra-train", line 33, in <module>
    sys.exit(load_entry_point('fairseq', 'console_scripts', 'fairseq-hydra-train')())
  File "/home/zhuozhu.zz/tops/fairseq/bin/fairseq-hydra-train", line 25, in importlib_load_entry_point
    return next(matches).load()
  File "/home/zhuozhu.zz/.local/lib/python3.7/site-packages/importlib_metadata/__init__.py", line 166, in load
    module = import_module(match.group('module'))
  File "/usr/local/lib/python3.7/importlib/__init__.py", line 127, in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
  File "<frozen importlib._bootstrap>", line 1006, in _gcd_import
  File "<frozen importlib._bootstrap>", line 983, in _find_and_load
  File "<frozen importlib._bootstrap>", line 967, in _find_and_load_unlocked
  File "<frozen importlib._bootstrap>", line 677, in _load_unlocked
  File "<frozen importlib._bootstrap_external>", line 728, in exec_module
  File "<frozen importlib._bootstrap>", line 219, in _call_with_frames_removed
  File "/home/zhuozhu.zz/fairseq/fairseq_cli/hydra_train.py", line 15, in <module>
    from fairseq import distributed_utils, metrics
  File "/home/zhuozhu.zz/fairseq/fairseq/__init__.py", line 21, in <module>
    from fairseq.logging import meters, metrics, progress_bar  # noqa
  File "/home/zhuozhu.zz/fairseq/fairseq/logging/progress_bar.py", line 315, in <module>
    from torch.utils.tensorboard import SummaryWriter
  File "/usr/local/lib/python3.7/site-packages/torch/utils/tensorboard/__init__.py", line 4, in <module>
    LooseVersion = distutils.version.LooseVersion
AttributeError: module 'distutils' has no attribute 'version'

解决方案：

$vim /usr/local/lib/python3.7/site-packages/torch/utils/tensorboard/__init__.py

import tensorboard
from packaging.version import Version as LooseVersion


```

```
    if not hasattr(tensorboard, '__version__') or LooseVersion(tensorboard.__version__) < LooseVersion('1.15'):
        raise ImportError('TensorBoard logging requires TensorBoard version 1.15 or above')

    del LooseVersion
    del tensorboard

    from .writer import FileWriter, SummaryWriter  # noqa: F401
    from tensorboard.summary.writer.record_writer import RecordWriter  # noqa: F401
```

Plain Text 复制代码

```
pip install --prefix=/home/zhuozhu.zz/tops/fairseq --editable ./
通过--prefix指定安装目录
```

Plain Text 复制代码

```
ValueError: numpy.ndarray size changed, may indicate binary incompatibility.
Expected 96 from C header, got 80 from PyObject

原因numpy 版本太低，解决方案 pip install numpy --upgrade
```

## 二、环境编译

### base_100h.yaml

Plain Text | 复制代码

```
$cat examples/wav2vec/config/finetuning/base_100h.yaml
# @package _group_
common:
  fp16: false
  log_format: json
  log_interval: 200
checkpoint:
  no_epoch_checkpoints: true
  best_checkpoint_metric: wer
task:
  _name: audio_pretraining
  data: /home/zhuozhu.zz/fairseq/manifest/finetune
  normalize: false
  labels: ltr
dataset:
  num_workers: 3
  max_tokens: 800000
  skip_invalid_size_inputs_valid_test: true
  valid_subset: valid
distributed_training:
  ddp_backend: legacy_ddp
  distributed_world_size: 3
criterion:
  _name: ctc
  zero_infinity: true
optimization:
  max_update: 80000
  lr: [0.00003]
  sentence_avg: true
  update_freq: [4]
optimizer:
  _name: adam
  adam_betas: (0.9,0.98)
  adam_eps: 1e-08
lr_scheduler:
  _name: tri_stage
  phase_ratio: [0.1, 0.4, 0.5]
  final_lr_scale: 0.05
model:
  _name: wav2vec_ctc
  w2v_path: /home/zhuozhu.zz/fairseq/outputs/2021-02-25/10-14-58/checkpoin
ts/checkpoint_last.pt
  apply_mask: true
  mask_prob: 0.65
  mask_channel_prob: 0.5
```

```
  mask_channel_length: 64
  layerdrop: 0.1
  activation_dropout: 0.1
  feature_grad_mult: 0.0
  freeze_finetune_updates: 0
```

### boost 编译

Plain Text 复制代码

```
./bootstrap.sh --prefix=/disk4/zhuozhu.zz/tops/boost-1.66 --with-python=python
./b2 define=_GLIBCXX_USE_CXX11_ABI=1 install -j5
```

### kenlm 编译

git diff CMakeLists.txt

Plain Text 复制代码

```
SET(CMAKE_C_COMPILER "/usr/local/bin/gcc")
SET(CMAKE_CXX_COMPILER "/usr/local/bin/g++")

set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -std=c++11 -D_GLIBCXX_USE_CXX11_ABI=1")
set(CMAKE_CXX_STANDARD 11)

add_definitions(-D_GLIBCXX_USE_CXX11_ABI=1)

SET(BOOST_INCLUDEDIR "/disk4/zhuozhu.zz/tops/boost-1.66/include")
SET(BOOST_LIBRARYDIR "/disk4/zhuozhu.zz/tops/boost-1.66/lib")

 # We need boost
find_package(Boost 1.66.0 REQUIRED COMPONENTS
```

#### setup.py

Plain Text 复制代码

```
ARGS = ['-g', '-DNDEBUG', '-DKENLM_MAX_ORDER='+max_order, '-std=c++11']
```

$git diff bindings/python/setup.py

```
diff --git a/bindings/python/setup.py b/bindings/python/setup.py
-        cfg = "Debug" if self.debug else "Release"
+
+        cfg = "Debug"
+        #cfg = "Debug" if self.debug else "Release"
         build_args = ["--config", cfg]
```

```
diff --git a/cmake/FindMKL.cmake b/cmake/FindMKL.cmake
-            SET(_found_gomp true)
+            SET(_found_gomp false)
             FOREACH(_lib_name ${OpenMP_CXX_LIB_NAMES})
```

## 三、学习资料

VQ-WAV2VEC: SELF-SUPERVISED LEARNING OF DISCRETE SPEECH REPRESENTATIONS
https://arxiv.org/pdf/1910.05453.pdf

wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations
https://arxiv.org/pdf/2006.11477.pdf

UniSpeech: Unified Speech Representation Learning with Labeled and Unlabeled Data
https://arxiv.org/pdf/2101.07597.pdf

Pushing the Limits of Semi-Supervised Learning for Automatic Speech Recognition
https://arxiv.org/pdf/2010.10504.pdf

Self-training and Pre-training are Complementary for Speech Recognition (Xu et al., 2020)
https://arxiv.org/pdf/2010.11430.pdf

fairseq 文档
https://fairseq.readthedocs.io/en/latest/data.html

wav2vec Librispeech

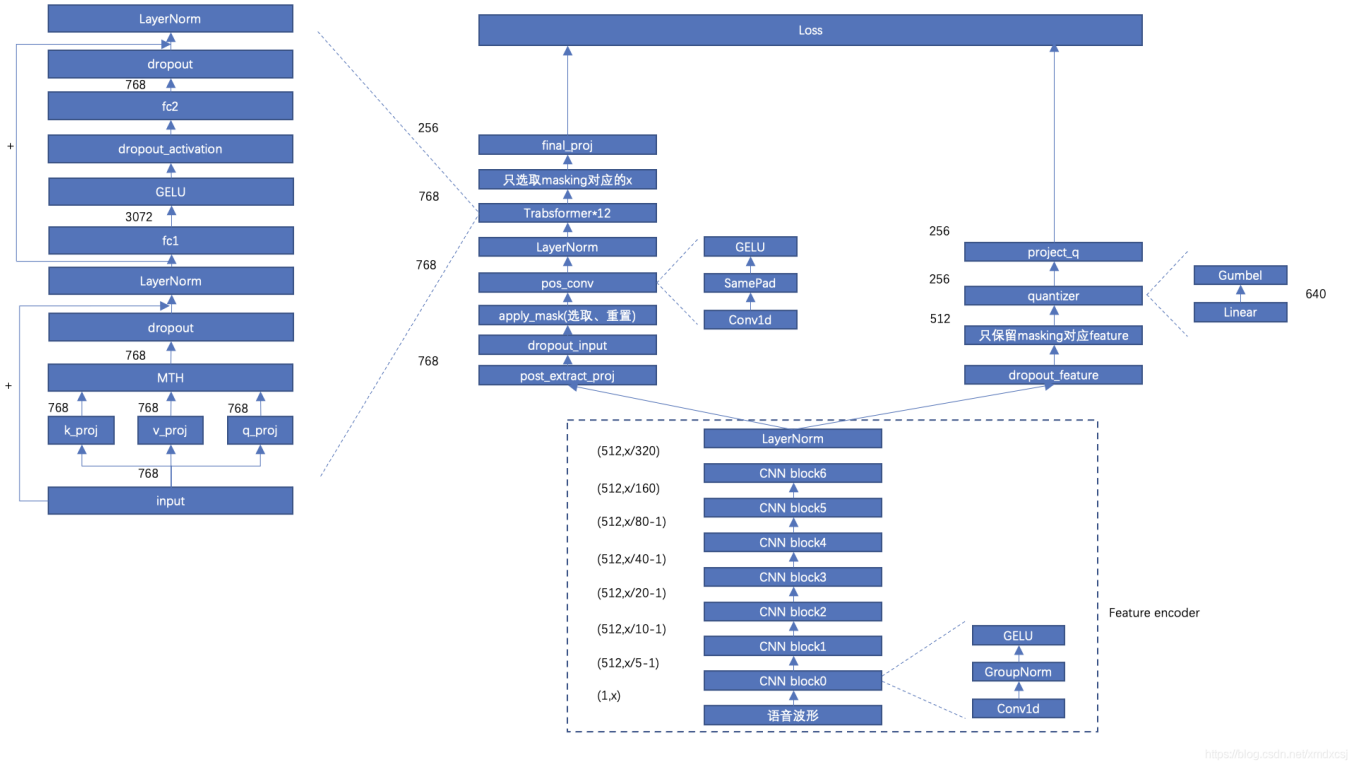
Plain Text | 复制代码

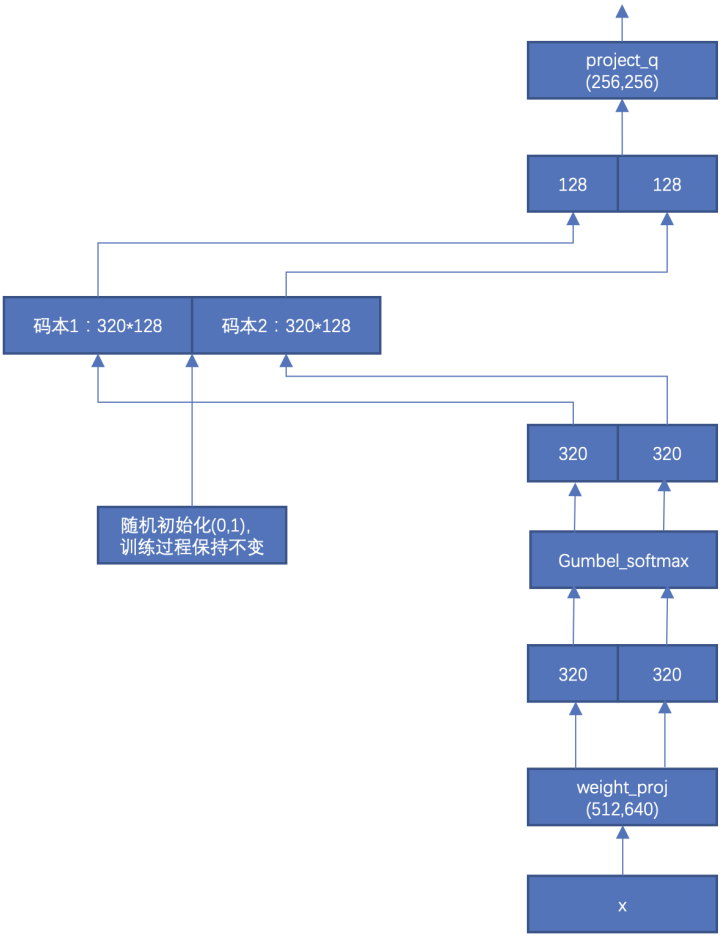
```
$sh run_librispeech_predict.sh
INFO:__main__:Namespace(all_gather_list_size=16384, autoregressive=False, azureml_logging=False, batch_size=None, batch_size_valid=None, beam=5, beam_size_token=100, beam_threshold=25.0, best_checkpoint_metric='loss', bf16=False, bpe=None, broadcast_buffers=False, bucket_cap_mb=25, checkpoint_shard_count=1, checkpoint_suffix='', constraints=None, cpu=False, criterion='ctc', curriculum=0, data='/disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/librispeech_wav2vec_test', data_buffer_size=10, dataset_impl=None, ddp_backend='pytorch_ddp', decoding_format=None, device_id=0, disable_validation=False, distributed_backend='nccl', distributed_init_method=None, distributed_no_spawn=False, distributed_port=-1, distributed_rank=0, distributed_world_size=1, diverse_beam_groups=-1, diverse_beam_strength=0.5, diversity_rate=-1.0, dump_emissions=None, dump_features=None, empty_cache_freq=0, enable_padding=False, eos=2, eval_wer=False, eval_wer_post_process='letter', eval_wer_tokenizer=None, fast_stat_sync=False, find_unused_parameters=False, finetune_from_model=None, fix_batches_to_gpus=False, fixed_validation_seed=None, force_anneal=None, fp16=False, fp16_init_scale=128, fp16_no_flatten_grads=False, fp16_scale_tolerance=0.0, fp16_scale_window=None, gen_subset='/disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/librispeech_wav2vec_test/train', heartbeat_timeout=-1, iter_decode_eos_penalty=0.0, iter_decode_force_max_iter=False, iter_decode_max_iter=10, iter_decode_with_beam=1, iter_decode_with_external_reranker=False, keep_best_checkpoints=-1, keep_interval_updates=-1, keep_last_epochs=-1, kenlm_model='/disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/lm.bin', kspmodel=None, labels='ltr', lenpen=1, lexicon='/disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/lexicon.txt', lm_path=None, lm_weight=2.0, load_checkpoint_on_all_dp_ranks=False, load_emissions=None, localsgd_frequency=3, log_format=None, log_interval=100, lr_scheduler='fixed', lr_shrink=0.1, match_source_len=False, max_len_a=0, max_len_b=200, max_sample_size=None, max_tokens=4000000, max_tokens_valid=4000000, maximize_best_checkpoint_metric=False, memory_efficient_bf16=False, memory_efficient_fp16=False, min_len=1, min_loss_scale=0.0001, min_sample_size=None, model_overrides='{}', model_parallel_size=1, nbest=1, no_beamable_mm=False, no_early_stop=False, no_epoch_checkpoints=False, no_last_checkpoints=False, no_progress_bar=False, no_repeat_ngram_size=0, no_save=False, no_save_optimizer_state=False, no_seed_provided=False, normalize=False, nprocs_per_node=1, num_shards=1, num_workers=1, optimizer=None, optimizer_overrides='{}', pad=1, path='/disk4/zhuozhu.zz/wav2vec_vox_960h_pl.pt', patience=-1, pipeline_balance=None, pipeline_checkpoint='never', pipeline_chunks=0, pipeline_decoder_balance=None, pipeline_decoder_devices=None, pipeline_devices=None, pipeline_encoder_balance=None, pipeline_encoder_devices=None, pipeline_model_parallel=False, post_process='letter', prefix_size=0, print_alignment=None, print_step=False, profile=False, quantization_config_path=None, quiet=False, replace_unk=None, required_batch_size_multiple=8, required_seq_len_multiple=1, reset_dataloader=False, reset_logging=False, reset_lr_scheduler=False, reset_meters=False, reset_optimizer=False, restor
```

```
e_file='checkpoint_last.pt', results_path='train', retain_dropout=False, r
etain_dropout_modules=None, retain_iter_history=False, rnnt_decoding_type
='greedy', rnnt_len_penalty=-0.5, sacrebleu=False, sample_rate=16000, samp
ling=False, sampling_topk=-1, sampling_topp=-1.0, save_dir='checkpoints',
save_interval=1, save_interval_updates=0, score_reference=False, scoring
='bleu', seed=1, shard_id=0, sil_weight=0.0, skip_invalid_size_inputs_vali
d_test=False, slowmo_algorithm='LocalSGD', slowmo_momentum=None, suppress_
crashes=False, task='audio_pretraining', temperature=1.0, tensorboard_logd
ir=None, threshold_loss_scale=None, tokenizer=None, tpu=False, train_subse
t='train', unit_lm=False, unk=3, unk_weight=-inf, unkpen=0, unnormalized=F
alse, user_dir=None, valid_subset='valid', validate_after_updates=0, valid
ate_interval=1, validate_interval_updates=0, w2l_decoder='kenlm', wandb_pr
oject=None, warmup_updates=0, wer_args=None, wer_kenlm_model=None, wer_lex
icon=None, wer_lm_weight=2.0, wer_word_score=-1.0, wfstlm=None, word_score
=-1.0, zero_infinity=False, zero_sharding='none')
INFO:__main__:| decoding with criterion ctc
dict_path /disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/librispeech_wav2ve
c_test/dict.ltr.txt
target_dictionary <fairseq.data.dictionary.Dictionary object at 0x7f579881
86a0>
INFO:__main__:| loading model(s) from /disk4/zhuozhu.zz/wav2vec_vox_960h_p
l.pt
INFO:fairseq.data.audio.raw_audio_dataset:loaded 2489, skipped 0 samples
tgt_dict ['<s>', '<pad>', '</s>', '<unk>', '|', 'E', 'T', 'A', 'O', 'N',
'I', 'H', 'S', 'R', 'D', 'L', 'U', 'M', 'W', 'C', 'F', 'G', 'Y', 'P',
'B', 'V', 'K', "'", 'X', 'J', 'Q', 'Z']
tgt_dict {'<s>': 0, '<pad>': 1, '</s>': 2, '<unk>': 3, '|': 4, 'E': 5,
'T': 6, 'A': 7, 'O': 8, 'N': 9, 'I': 10, 'H': 11, 'S': 12, 'R': 13, 'D': 1
4, 'L': 15, 'U': 16, 'M': 17, 'W': 18, 'C': 19, 'F': 20, 'G': 21, 'Y': 2
2, 'P': 23, 'B': 24, 'V': 25, 'K': 26, "'": 27, 'X': 28, 'J': 29, 'Q': 3
0, 'Z': 31}
INFO:__main__:| /disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/librispeech_
wav2vec_test /disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/librispeech_wav
2vec_test/train 2489 examples
<fairseq.data.add_target_dataset.AddTargetDataset object at 0x7f57954a1c18
>
use W2lKenLMDecoder
[flashlight] load LM start.
[flashlight] load LM finish.
load vocab success.
token HAMMER,lmIdx 11030
token TO,lmIdx 11707
token HANDLE,lmIdx 5552
token CONTRAST,lmIdx 10751
token TREMENDOUSLY,lmIdx 13803
token ARGUING,lmIdx 12788
token THORLEIF,lmIdx 4043
token COVERING,lmIdx 8529
```

```
token WAVES,lmIdx 8381
token EXERTED,lmIdx 14503
token PLACARD,lmIdx 3869
INFO:__main__:WER: 1.4511028381569993
INFO:__main__:| Processed 2489 sentences (273750 tokens) in 3445.1s (0.72s
entences/s, 79.46 tokens/s)
INFO:__main__:| Generate /disk4/zhuozhu.zz/librispeech_raw/LibriSpeech/lib
rispeech_wav2vec_test/train with beam=5
```

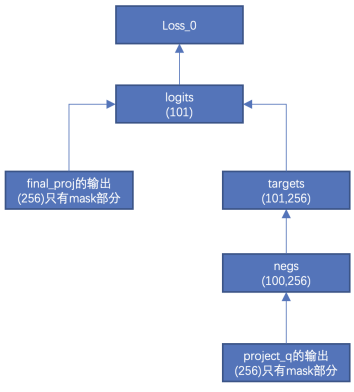
https://blog.csdn.net/xmdxcsj/article/details/115787729






两个向量中为1的idx为i和j，从码本1提取第i行，从码本2提取第j行，然后拼接

320维的向量，最大值对应的idx设为1，其他为0

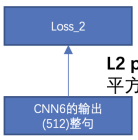



F.cross_entropy
求CE，得到**Contrastive Loss**

compute_preds()
1.将y和negs拼接得到target，y（即idx=0）表示x对应的groud truth
2.target和x分别求cosine距离，期望idx=0的位置距离最小，即x和y距离最近，跟另外100个负样本距离远

sample_negatives()
从同一句话的所有mask帧里面选取100个作为负样本


**Diversity Loss:**
\sum p*logp




**L2 penalty**：
平方求均值

loss部分