FACEBOOK AI

Joint Masked CPC and CTC Training for ASR

Chaitanya Talnikar,
Likhomanenko Tatiana,
Ronan Collobert,
Gabriel Synnaeve

Facebook AI Research

4384
IEEE ICASSP 2021

Overview

Self-supervised training for ASR requires two stages:

- pre-training on unlabeled data;
- fine-tuning on labeled data.

We propose **joint training:**

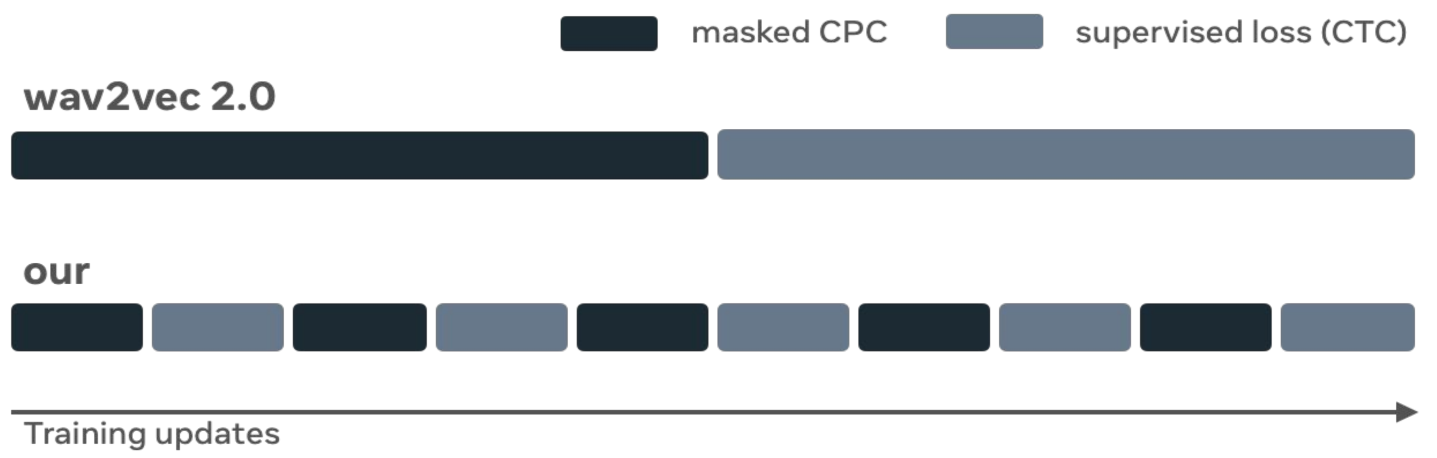
- **alternate** supervised and unsupervised losses minimization, thus directly optimize for ASR task rather than for unsupervised task.

Result:

- **simplified** learning process that **matches** state-of-the-art two-stage pipeline.

Joint Training

We jointly minimize two losses, a supervised L_s and an unsupervised L_u, by alternating between minimizing L_s on labeled data and minimizing L_u on unlabeled data.



Model: takes input raw audio $\boldsymbol{x}$ and outputs token $\boldsymbol{y}$ probabilities at time t

$$\boldsymbol{z} = f(\boldsymbol{x}) \quad (1) \quad \text{convolutional encoder}$$

$$\tilde{\boldsymbol{z}} = g(\text{mask}(\boldsymbol{z})) \quad (2) \quad \text{transformer context network}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x}) = h(\tilde{\boldsymbol{z}}). \quad (3)$$

Supervised loss: Connectionist Temporal Classification (CTC).

Unsupervised loss: wav2vec 2.0 self-supervision loss; can be viewed as a contrastive predictive coding (CPC) loss where the task is to predict the masked encoder features.

$$\mathcal{L}_u(\boldsymbol{\theta}, \boldsymbol{x}) = \frac{1}{T}\underbrace{\sum_t}_{\text{masked positions}} -\log \frac{s(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t)}{s(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t) + \underbrace{\sum_{t'} s(\boldsymbol{z}_{t'}, \tilde{\boldsymbol{z}}_t)}_{\text{non-masked positions}}} \quad (4) \qquad s(\boldsymbol{z}_t, \tilde{\boldsymbol{z}}_t) = \frac{1}{\tau}\exp\left(\frac{\boldsymbol{z}_t \cdot \tilde{\boldsymbol{z}}_t}{\|\boldsymbol{z}_t\|\|\tilde{\boldsymbol{z}}_t\|}\right)$$

Alternate minimization:

separate adaptive momentum optimizers are used for each of the two losses with different learning rates η_s and η_u;

optimizers maintain their state independently, while sharing the model parameters.

Algorithm 1: Alternating minimization algorithm.

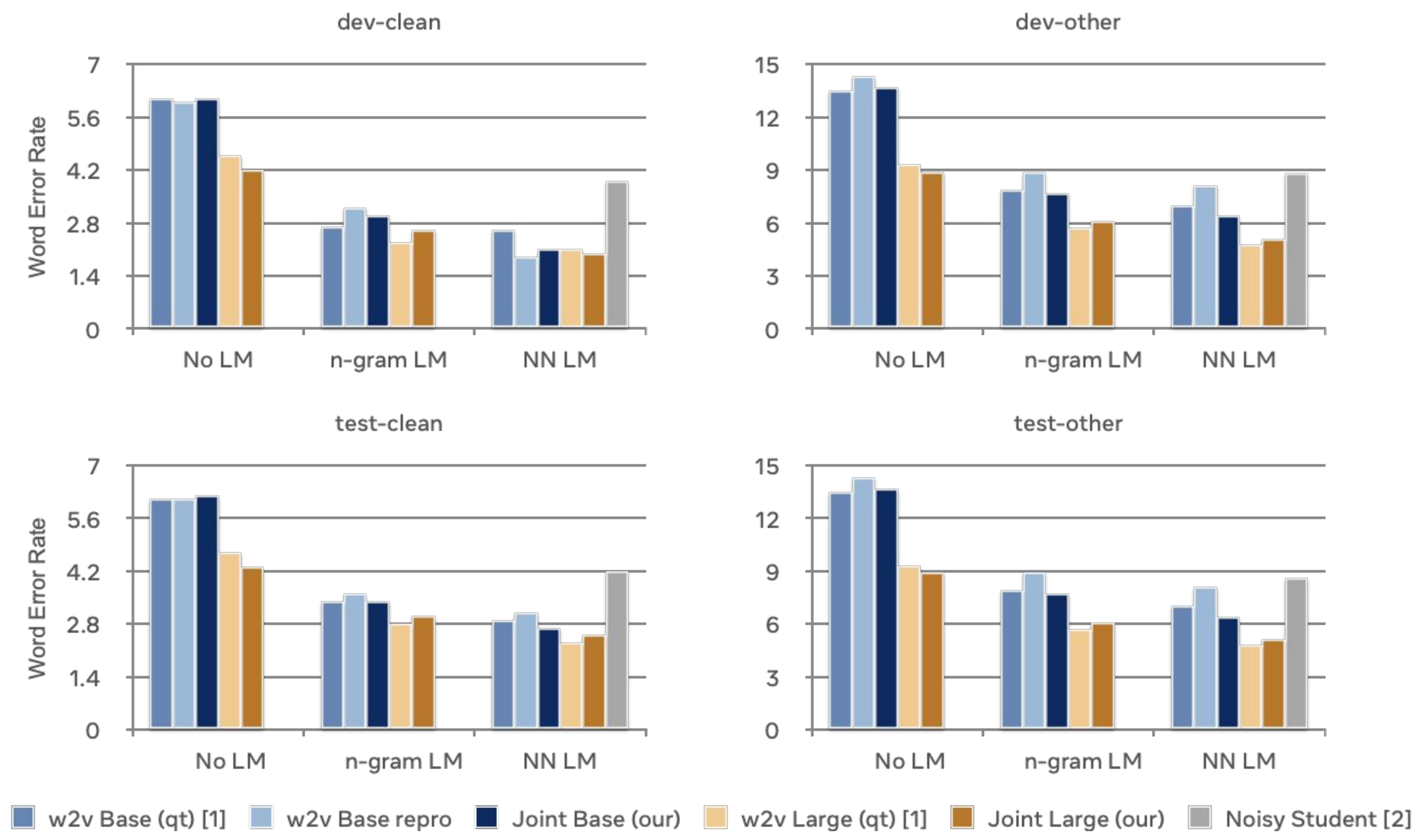
Data: Labeled data $L = \{\boldsymbol{x}, \boldsymbol{y}\}$, Unlabeled data $U = \{\boldsymbol{x}\}$
Result: Acoustic model $p_{\boldsymbol{\theta}}$
Randomly initialize parameters of the acoustic model $p_{\boldsymbol{\theta}}$;
repeat
  repeat
    1. Forward the model with Eq. (1) and (2) obtaining $\boldsymbol{z}$ and $\tilde{\boldsymbol{z}}$
    2. Compute $g_u = \nabla_{\boldsymbol{\theta}}\mathcal{L}_u(\boldsymbol{\theta}, \boldsymbol{x})$ using $\boldsymbol{z}$, $\tilde{\boldsymbol{z}}$
    3. Update $p_{\boldsymbol{\theta}}$ with η_u and g_u
  until *N times for $\boldsymbol{x} \in U$*;
  4. Forward the model for $\boldsymbol{x} \in L$ with Eq. (1)-(3) obtaining $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$
  5. Compute $g_s = \nabla_{\boldsymbol{\theta}}\mathcal{L}_s(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$ using $p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$
  6. Update $p_{\boldsymbol{\theta}}$ with η_s and g_s
until *convergence in word error rate or maximum iterations are reached*;

Experimental Setup

- **Data**: i) 960h of LibriSpeech is used as unlabeled set; ii) 100h of *train-clean* LibriSpeech is used as labeled.
- **Models**: *Base* (94M) and *Large* (315M) wav2vec 2.0 architectures consisting of convolutional encoder, transformer context network
- **Tokens**: English alphabet.
- **Data augmentation in the ASR task**: a variation of SpecAugment that uses the same masking procedure as the contrastive loss
- **Training**: 500k updates with Adam optimizer.

Results

- Joint training **matches** the word error rates (WER) of the wav2vec 2.0 for either model architecture (*Base* and *Large*) on both sets (*test-clean* and *test-other*), with and without a language model (LM).
- Unlike the wav2vec 2.0 model, our model is quantization-free, operates in the continuous space and does not use any unsupervised loss penalty terms during training.



Effect of Hyperparameters on Downstream Task

- Training is not sensitive to the number L_u to L_s updates.
- Lower L_u to L_s learning rate ratio or a single optimizer results in a higher word error rate.

Word error rate (*dev-other*, 4-gram LM) of models with different hyperparameters compared to baseline.

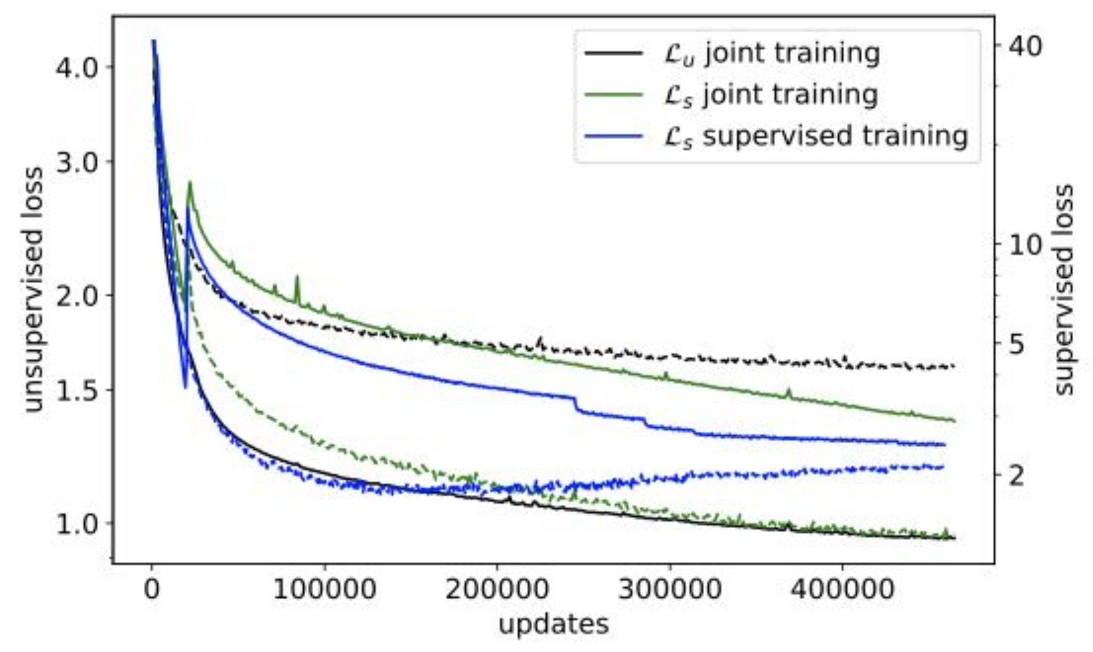
Hyperparameter	Updates	LR	dev-other WER
Baseline	1:1	20:1	8.0
L_u to L_s update ratio	5:1	20:1	7.9
L_u to L_s learning rate ratio	1:1	4:1	9.0
Single optimizer	1:1	20:1	11.1

Regularization Effect on Supervised Loss

Observations suggest **regularizing** effect to the supervised loss:

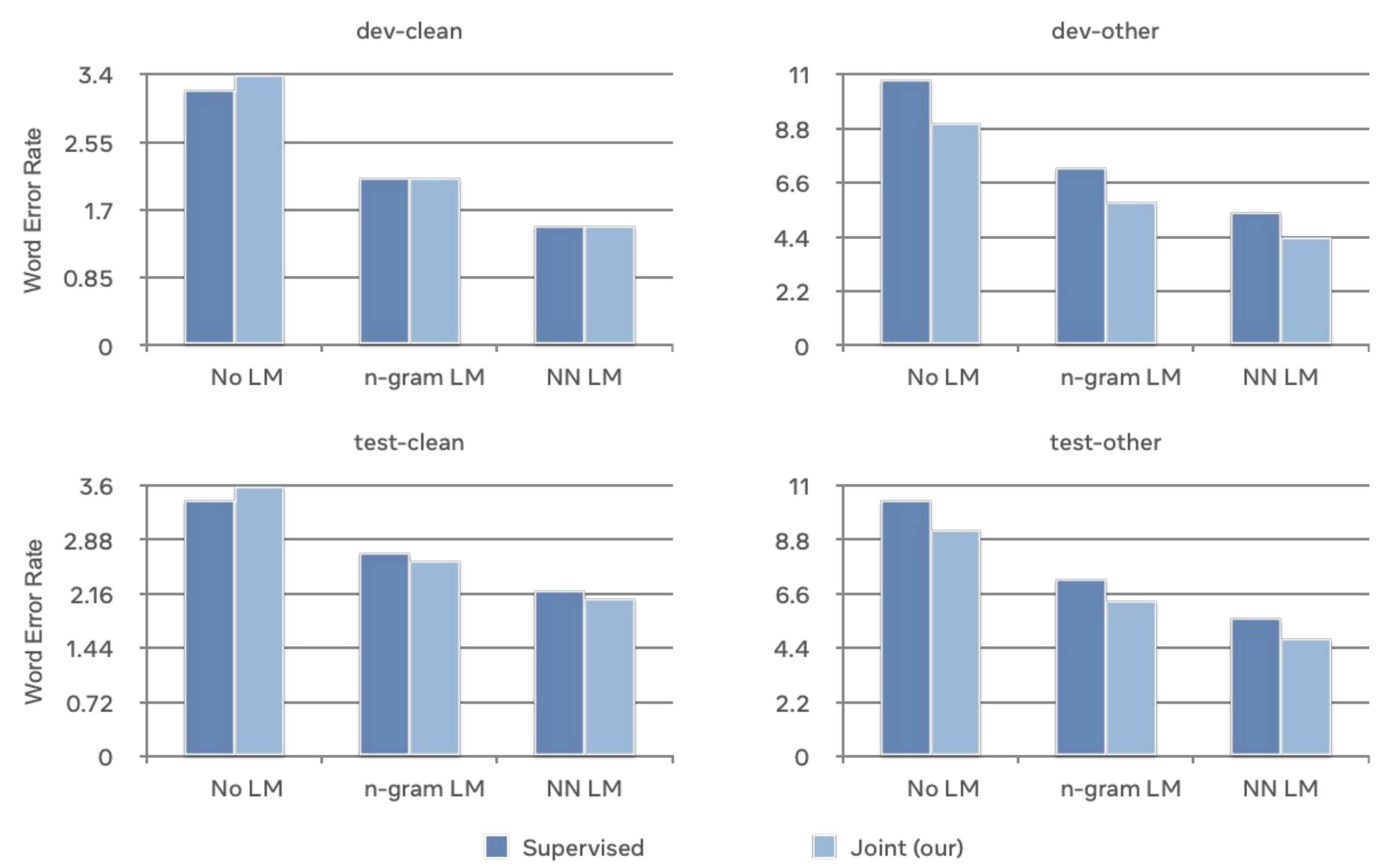
- joint training achieves **lower** supervised loss on the **validation** and a **higher** supervised loss on the **train** compared to supervised training.

Unsupervised L_u and supervised L_s losses behaviour on the train (solid) and validation (dotted) sets for joining training (L_u black and L_s green) and supervised only training (L_s blue). All 960h are used with labels.



- lower WER compared to a supervised model (despite lower number of updates from supervised loss).

Word error rates of models trained on 960h of LibriSpeech (all 960h are used with labels).



Acknowledgement

We would like to thank Alexei Baevski and Michael Auli for helpful discussions regarding wav2vec 2.0.

References

[1] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," Advances in Neural Information Processing Systems, vol. 33, 2020.

[2] Daniel S Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V Le, "Improved noisy student training for automatic speech recognition," Proc. Interspeech 2020, pp. 2817–2821, 2020.