EasyChair Preprint

№ 11660

Vietnamese Automatic Speech Recognition: Self-Supervised and Semi-Supervised Learning Techniques Combination

Duong Trinh

January 2, 2024

Vietnamese Automatic Speech Recognition: Self-Supervised and Semi-Supervised Learning Techniques Combination

Anh-Duong Trinh
VCCorp / Hanoi, Vietnam
anhduong00799@gmail.com

Abstract

The speech recognition task in Vietnamese is increasingly being interested and invested in by researchers and organizations. With a small amount of training data, self-supervised models have performed better than supervised models in speech recognition. As a part of this study, I explored two different learning methods, self-supervised learning and semi-supervised learning, in combination to solve the speech recognition problem. In order to perform self-supervised learning, I use a HuBERT model, which combines offline clustering with a BERT-like prediction loss. On the HuBERT model, I use the Gradient Mask technique to perform semi-supervised learning. Approximately 500 hours of unlabeled data and 50 hours of labeled data are provided by the VLSP 2022 organizers for training. The approach performs third on the ASR-T1 test using the proposed methodology, with a Syllable Error Rate (SyER) of 14.28%.

Index Terms: speech recognition, self-supervised learning, pseudo-labeling

1 Introduction

The collecting of huge amounts of labeled data for speech recognition requires a significant amount of time and effort. To solve the problem of labeled data, self-supervised speech recognition algorithms have recently become popular. Self-supervised learning (SSL) is a rapidly developing subclass of unsupervised learning systems that employ information collected from the input data itself as the label to learn representations helpful for downstream tasks.

S. Schneider et al. [1] improve supervised speech recognition using unsupervised pre-training in wav2vec. The wav2vec model is a convolutional neural network that uses raw audio to build a representation that can be input into a speech recognition system. The objective is to obtain a contrastive loss by discriminating between actual future audio samples and negatives. Later, Wav2vec 2.0 was proposed by A. Baevski [2]. When solving a contrastive task based on a quantization of the jointly acquired latent representations, wav2vec 2.0 masks the speech input in the latent space. The model is trained using a contrastive task in which the true latent is to be differentiated from distractions. The latent representations are supplied to a Transformer network to produce contextualized representations. Another strategy uses an offline clustering phase to produce aligned target labels for a prediction loss that is similar to BERT. The Hidden-Unit BERT (HuBERT) approach for learning self-supervised speech representation was proposed by Wei-Ning Hsu et al. [3].

Semi-supervised learning bridges supervised learning and unsupervised learning techniques to solve their key challenges. With it, you train an initial model on a few labeled samples and then iteratively apply it to the greater number of unlabeled data. To provide a learning signal to a discriminative model trained on unlabeled speech, Wei-Ning Hsu et al. [4] propose local prior matching (LPM), a semi-supervised objective that extracts knowledge from a strong prior (such as a language model). A proposal model first creates a set of hypotheses from an unlabeled utterance. The ASR model can then incorporate prior information by distilling it using the language model's (LM) target distribution. Self-training using WFST-based supervision is conducted using the recently proposed graph-based temporal classification (GTC) objective by Peter et al. [5], which is derived from an N-best set of pseudo-labels. Shaoshi Ling proposes the Gradient Mask [6] to enhance pseudo-label training in end-to-end speech recognition by drawing inspiration from masked prediction and incorporating its idea. The ASR model uses the Gradient Mask to train a student model on the pseudo labels

after first training a seed model to generate labels.

In this paper, I propose a method for solving speech recognition tasks in The 9th International Workshop on Vietnamese Language and Speech Processing (VLSP2022) competition by combining Hubert model with Gradient Mask technique. To learn the speech representation, I first train the HuBERT model using labeled and unlabeled data that the competition organizers provided. Approximately 50 hours of labeled audio data are then used to fine-tune the HuBERT model. I produce pseudo labels from the fine-tuned HuBERT model and then use the Gradient Mask to train a student model on the pseudo labels. By masking the gradients related to unmasked input, the model only allows gradients corresponding to masked input to back-propagate through the model encoder. The model is trained by minimizing the loss on both labeled and pseudo-label data while turning off the Gradient Mask on labeled data.

The training method can force the model to learn a strong acoustic representation. Intuitively, the HuBERT model is required to learn both acoustic and language models from continuous inputs. First, the model must transform unmasked inputs into useful continuous latent representations, which corresponds to the classical acoustic modeling challenge. Second, the model must capture the long-range temporal relationships between learned representations in order to reduce prediction error [3]. Additionally, by reducing the impact of label noise on the model, gradient mask can enhance pseudo-label training.

2 Method

2.1 HuBERT model

Unlike Discrete BERT [7], which relies on an advanced representation learning model to discretize continuous inputs, the Hidden Unit BERT (HuBERT) approach uses quantized MFCC features as targets learned with classic k-means. The k-means model $g_1(.)$ thus assigns a cluster center to every timestep in order to compute the targets. The HuBERT model architecture is based on the wav2vec 2.0 architecture [8], with a convolutional module $f_1(.)$ and a Transformer encoder $f_2(.)$, as well as a softmax normalized output layer $g_2(.)$:

$$c_t = g_1(X_{[t-w,t+w]}) \tag{1}$$

$$z_t = f_1(X_{[t-u,t+u]}) \tag{2}$$

$$H = f_2(m(Z)) \tag{3}$$

$$C_t = g_2(h_t) \tag{4}$$

where w defines the window size used to compute the MFCCs features. Both masked, L_m, and unmasked, L_u, timesteps are used in the computation of the categorical cross-entropy loss:

$$L_m = \sum_{t \in M} -logp(c_t|X) \tag{5}$$

$$L = \beta L_m + (1-\beta)L_u \tag{6}$$

Again, M is the set of all masked timesteps, β is a scalar hyperparameter and L_u is computed as L_m but summing over $t \notin M$. The significance of target consistency, which enables the model to concentrate on modeling the input's sequential structure, is a key insight that motivates this work. Importantly, pre-training is a two-step process for HuBERT. These two steps are pseudo-label generation and speech representation learning.

2.2 Gradient mask

The gradient mask approach has been researched to effectively use the unlabeled data source in order to benefit from unlabeled data in the same domain that is less affected by incorrect pseudo labels [6] [9]. For sequence $X = [x_1, ..., x_T]$ which has pseudo labels $Y^{'} = [y_1^{'}, ..., y_u^{'}]$. With the objective of allowing the model to predict labels from masked features, the ASR model has been trained to be a robust acoustic representation model that can benefit in ASR tasks. For input sequence Z, $mask = [m_1, ..., m_T]$, I randomly generated a mask representing the positions of the hidden features before feeding them to the Transformer encoder. Specifically, m_t is 1 if features are masked at time t, and 0 otherwise. A learned mask embedding emb replaces hidden features in masked positions. The following is a representation of the encoder features function:

$$h^{enc} = f_{enc}((\sim mask) * f_1(X) + mask * emb) \tag{7}$$

The mask strategy is the same as [2], where I randomly select p starting indices from all time steps to serve as samples without replacement and use overlap spans to mask the remaining m time steps from each sampled index. I used a mask

sequence to mask the gradients corresponding to the unmasked inputs when the gradient is back-propagated to the encoder:

$$grad_{h^{enc}} = (\sim mask) * grad_{h^{enc}} \quad (8)$$

Figure 1 presents an illustration of the training model.

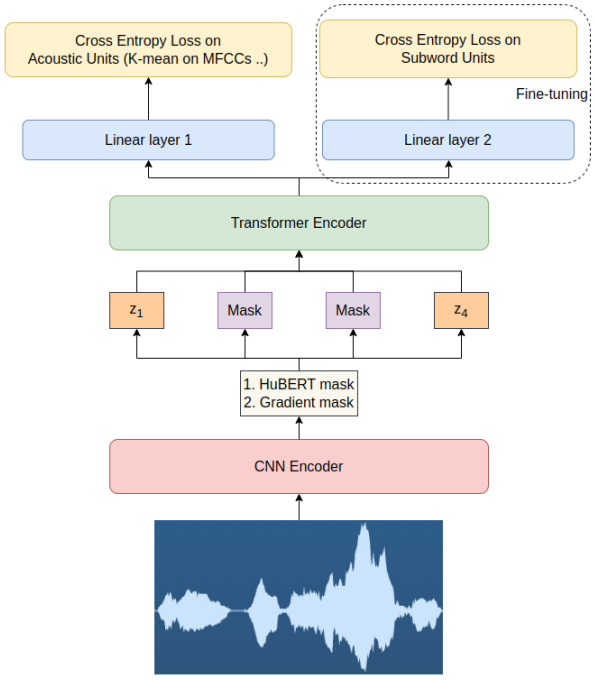

Figure 1: The HuBERT mask method was used for pretraining, while the Gradient mask method was used for fine-tuning using pseudo-label.

3 Experiments

3.1 Data

The total 550 hours of VLSP 2022 task 01 audio are used for unsupervised pre-training. The audio dataset for supervised fine-tuning contains about 50 hours of labeled audio data. Audio segments longer than 18 seconds and shorter than 0.8 seconds are removed.

I use 3096 subword [10] units as my prediction targets. I did not apply any audio data augmentation methods when training on labeled and pseudo-label data.

3.2 Training procedure

3.2.1 Pre-Training

On the 550 hours of VLSP 2022 task 01 audio on 1 GPU, I train the HuBERT BASE model for three iterations. The first iteration is trained for 400k steps, and the second iteration is trained for 800k steps using labels produced by clustering the output of the first iteration model's sixth transformer layer. The third iteration is similar to the second iteration; however, the model is trained over 1.2M steps using labels generated by clustering the output of the second iteration model's ninth transformer layer.

I perform k-means clustering with 100 clusters on 39 dimensional MFCC features, which are 13 coefficients with the first and second-order derivatives, to produce labels for the first iteration HuBERT training over the VLSP 2022 task 01 training set. I run k-means clustering with 500 clusters on the latent features extracted from the HuBERT model pre-trained in the previous iteration (not fine-tuned) at some intermediate transformer layer to generate better targets for following iterations.

Mask span is set to $l = 10$ for all HuBERT configurations, and $p = 8\%$ of the convolutional module output frames are randomly selected as mask start. The learning rate ramps up linearly from zero to the peak learning rate for the first 8% of the training steps, then decays linearly down to zero when the Adam [11] optimizer is used with $\beta = (0.9, 0.98)$. For the HuBERT BASE model, the peak learning rate is 5e-4.

3.2.2 Supervised Fine-Tuning and Decoding

I fine-tune the HuBERT BASE model on a single GPU using 50 hours of annotated audio data. The convolutional module audio encoder parameters are fixed during fine-tuning. The model training uses a freeze-step hyperparameter, similar to wav2vec2.0, to determine how many fine-tuning steps the transformer parameters are fixed, and only the new softmax matrix is trained. For the first 30% of the training steps, the learning rate gradually increases from 1e-5 to 1e-4 before decreasing linearly to 0.

At the time of inference, log-linear interpolation is used to incorporate the external language model. I utilize a technique called shallow fusion, which combines language modeling with beam search decoding techniques [12]:

$$Score = logP_{CTC}(Y|X) + \alpha logP_{LM}(Y) + \beta|Y| \quad (9)$$

where Y is the predicted text, $|Y|$ is number word in text, and β and α stand for the word score and language model weight, respectively. I use the 5-gram language model in all of my decoding experiments [13].

Table 1: WER on the VLSP 2022 Test set 01 for the ASR system

Models	Data	WER
HuBERT BASE	50-hours labeled	18.89
HuBERT BASE GM-01	50-hours labeled and 500h pseudo-labeled	14.52
HuBERT BASE GM-02	50-hours labeled and 500h pseudo-labeled	14.28

3.2.3 Semi-Supervised Training

Let $L = \{x_i, y_i\}$ be a labeled dataset and $U = \{x_j\}$ be a large unlabeled dataset. I use a fine-tuned HuBERT BASE model on 50 hours of labeled audio data to generate pseudo-labeled on dataset $U = \{x_j, y_j^{'}\}$. The following step is to train a student model with both datasets L and U. I use the gradient mask method as described in 2.2 to update the model parameters using a minibatch from the pseudo-labels dataset U. I update parameters in the standard way on a minibatch from the labeled dataset L. The pseudo-label generation process is repeated twice.

3.3 Results

The Syllable Error Rate (SyER) metric will be used to evaluate the models' quality:

$$SyER = \frac{S+D+I}{N} \tag{10}$$

where: S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct syllables, N is the number of syllables in the reference $N = S + D + C$

After training the HuBERT BASE model before and after the gradient mask was used, Table 1 shows the WER results on the VLSP 2022 task 01 test set. After pre-training, the HuBERT BASE model is fine tuned on a labeled dataset. The models HuBERT BASE GM-01 and HuBERT BASE GM-02 use the gradient mask approach.

The table shows the results of the student model directly trained from the pseudo labels given by the HuBERT BASE model for HuBERT BASE GM-01. The Syllable Error Rate (SyER) for the test set decreased by 4% when the gradient mask technique was applied. I repeat the pseudo labeling technique twice for HuBERT BASE GM-02.

Because the quality of the pseudo-label influences the ASR model quality in semi-supervised learning, I choose the hubert model to generate the pseudo-label for the first iteration. The CNN encoder layer's weights are not altered when training using pseudo-label, only the emd (mask embedding) and transformer layers are, reducing the effect of incorrect pseudo-label on the model. The weights of the transformer layers will also be modified when training with batches without a pseudo-label. The results of the experiments in this paper show that the Gradient Mask technique can be applied to the self-supervised HuBERT model, significantly boosting the accuracy of the ASR model while using only one GPU for training.

4 Conclusions

By combining self-supervised and semi-supervised methods, I propose a straightforward and effective technique for improving pseudo-label training. This method enables the model to learn the strong acoustic features while also being effective with label noise in pseudo-label training. In the future, Distributed Training on many GPUs will be used to improve the ASR model during the Pre-Training phase.

References

[1] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*, pages 3465–3469, 2019.

[2] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. Wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[3] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3451–3460, oct 2021.

[4] Wei-Ning Hsu, Ann Lee, Gabriel Synnaeve, and Awni Hannun. Semi-Supervised Speech Recognition via Local Prior Matching. *arXiv e-prints*, page arXiv:2002.10336, February 2020.

[5] Niko Moritz, Takaaki Hori, and Jonathan Le Roux. Semi-supervised speech recognition via graph-based temporal classification. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6548–6552, 2021.

[6] Shaoshi Ling, Chen Shen, Meng Cai, and Zejun Ma. Improving pseudo-label training for end-to-end speech recognition using gradient mask. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8397–8401, 2022.

[7] Alexei Baevski, Michael Auli, and Abdel rahman Mohamed. Effectiveness of self-supervised pre-training for speech recognition. *ArXiv*, abs/1911.03912, 2019.

[8] Abdel rahman Mohamed, Hung yi Lee, Lasse Borgholt, Jakob Drachmann Havtorn, Joakim Edin, C. Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16:1179–1210, 2022.

[9] Dang Dinh Son, Le Dang Linh, Dang Xuan Vuong, Duong Quang Tien, and Ta Bao Thang. Asr - vlsp 2021: Conformer with gradient mask and stochastic weight averaging for vietnamese automatic speech recognition. *VNU Journal of Science: Computer Science and Communication Engineering*, 38(1), 2022.

[10] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[12] Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *ArXiv*, abs/1503.03535, 2015.

[13] Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.