Joint CTC-Transformer (JCT) is an encoder-decoder structure in end-to-end speech recognition. Based on the structure of Transformer, acoustic features are applied as input, on top of the encoder, connectionist temporal classification (CTC) loss performs as the prediction target, decoder blocks remain unchanged, we call it JCT. In this paper, we propose a pre-trained method for the encoder of JCT to attain high-level representations of acoustic features, which leverages massive unlabeled audio data. Then these representations are applied to train entire JCT structure with a small amount of supervised data. We exploit bidirectional transformer to implement it and made a comparison with totally supervised JCT. All the experiments are conducted on WSJ audio corpus and librispeech corpus. After several trials, the two-stage training method deliver exceptional better performance than totally supervised model. Moreover, the word error rate with two-stage training which only exploits 30% of WSJ labeled data achieves 17% reduction than which trained by 50% of WSJ in a totally supervised way.

Index Terms: unsupervised learning, transformer, ASR

1. Introduction

Unsupervised learning [1] plays an important role in deep learning, especially because data labeling is quite time consuming and highly human cost. However, most existed automatic speech recognition systems are only based on substantial labeled data, and take no advantage of unlabeled data. In order to make use of unlabeled data, we propose a semi-supervised structure which combines unsupervised pre-training and supervised training together. Our unsupervised pre-training process is mainly inspired by the unsupervised pre-training process in natural language processing (NLP) tasks, especially the most representative work BERT [2] which has refreshed state-of-the-art of dozens of NLP tasks. Our supervised training structure JCT is inspired by Transformer [3] and CTC [4]. Transformer possess capable ability in parallel computing and long sequences modeling, it has been widely leveraged into end-to-end speech recognition, such as [5][6][7][8], which showed great superiority than recurrent neural network (RNN) based models [10][11]. CTC is an conventional end-to-end speech recognition loss function. In JCT model, CTC simply acts as an auxiliary function in supervised training process. Consequently, we exploit the encoder-decoder network JCT, which jointly trained CTC and transformer through a shared encoder during the supervised training process.

In this paper, we propose a masked pre-trained model structure like BERT and then fine-tuned it in JCT with multi task learning method. BERT is a pre-trained language model (LM) which consists of masked LM task and next sentence prediction task that captures the word level and sentence level representation, respectively. While for ASR tasks, due to lack of contextual coherence information in acoustic samples, we abandon next sentence prediction task in our pre-trained model. Meanwhile, in masked LM task, BERT generates masks for original text data with special mask token ([MASK]). However acoustic features such as Mel-Frequency Cepstral Coefficients (MFCC) features and log-mel filter bank (Fbank) features [12] are much more complex than plain text features, the unclear alignment between acoustic frames and their transcriptions make it impossible to mask raw audio data in semantic level. Naturally, we mask the frames in neural networks. The implementation structure of the pre-trained model is a deep bidirectional transformer. Figure 1 demonstrates the structure of masked pre-trained model. We use Fbank features as input and mask 15% of the input down-sampled frames. Different from the conventional approach, our pre-training process exploits the information from the past and future frames to establish present masked frame, frames are then reconstructed as context representations. As a kind of novel high-level represen-

Figure 1: The structure of masked pre-trained model
high-level acoustic representations. Wave2vec [13] proposed an unsupervised pre-training method by learning from the original audio signals rather than Fbank features, optimized by the noise contrastive estimation (NCE) of a binary classification task. Contrastive Predictive Coding (CPC) [14] dedicated to compress the higher-dimensional data into a more compact potentially embedded space where conditional prediction is easier to be modeled, then the researchers construct powerful autoregressive models in this potential space to make multi-step future predictions, CPC is also optimized by NCE. Compared with CPC, Autoregressive Predictive Coding (APC) [15] mainly focused on predicting the spectrum of a future frame rather than a wave sample, which appears like language model. The researchers use RNN based model to reconstruct future frame with information from its past frames, and the optimization is done by the reconstruction discrepancy.

Recently published literature deep contextualized acoustic representations (DeCoAR) [16] introduced a new representation learning method in which a temporal slice of filterbank features from past and future context frames are reconstructed, and is implemented by bi-directional LSTM networks and optimized by reconstruction error. Mockingjay [17] proposed a speech representation learning approach as BERT, where bidirectional Transformer encoders are pre-trained on a large amount of unlabeled speech data and these representations are applied to a wide range of downstream tasks in ASR. Unlike their work, we mask the frames after down sampling layer while Mockingjay directly masks Fbank features before down sampling layer, we also exploit different down-sampling method and distinctive supervised learning strategy from Mockingjay.

3. Semi-supervised JCT

In this section, firstly we introduce the details of unsupervised MPE and its optimization target, then we give the description of supervised JCT and its multi-task training approach. At last, we present the fine-tuning methods of applying pre-trained representations to supervised downstream tasks.

3.1. Unsupervised pre-trained encoder

Since CPC and wave2vec use autoregressive models to encode temporal information based on past acoustic sequence, which limits the potential of speech representation learning and decrease the training speed in pre-training stage. We leverage bi-directional transformer to reconstruct current masked frame through not only its past but future frames. The structure of MPE is illustrated on the left of Figure 2, which consists of three parts: down sampling layer, mask layer, bi-directional Transformer block. Considering the smoothness of acoustic spectrograms and faster calculation in training process, we place two convolutional layers before the transformer attention layer to exploit the structure locality of spectrograms [18]. We apply striding methods in both two layers, which down-sampled feature map to a quarter of the original length. After that, we add a linear projection layer to reshape the dimensions of features to fit for the input of the transformer. Then we present a random mask after the linear hidden layer with following rules: 15% of the input frames need to be masked. According to a uniform distribution, for every single frame within these selected frames: there’s a probability of 80% that this frame will get converted to 0 vector, a probability of 10% to be transformed as a random frame, a probability of 10% to remain unchanged. We also add sinusoidal positional embedding to the input features. The bi-directional Transformer block consists of $N_t$ layers of modules that can be stacked on top of each other multiple times. Each module composed of two sub-layers: multi-head attention layer and feed-forward layer, each sub-layer in each encoder has a residual connection around it, and is followed by a layer-normalization step. Given $t$ as length of input features, $T$ as length of MPE output sequences. $x = (x_1, x_2, ..., x_t)$, $e = (e_1, e_2, ..., e_T)$ respectively represent input features and reconstructed representations. $h = (h_1, h_2, ..., h_T)$ is the masked down-sampled acoustic features.

$$h = \text{Mask}(\text{Conv}(\text{Conv}(x)))$$

$$e = h + \text{SubBlock}(h)$$

Thus, the reconstruction discrepancy can be depicted as:

$$L_{pre} = \sum_{i=1}^{T} |h_i - e_i|$$

The element in loss function merely contains the frames that has been masked rather than those always keep unchanged.

3.2. Supervised encoder-decoder structure

We exploit JCT in downstream supervised tasks. Based on the encoder-decoder structure of Transformer, on top of the encoder, CTC loss has been added as the prediction target. Since pure CTC-based model always works together with a language model because of its independent assumption to the output elements. While pure data-driven attention-based model is hard to learn from scratch due to the sensitivity of attention mechanism. Consequently, we integrate CTC with Transformer through the shared encoder MPE. In our experiments, we found that attention mechanism tends to be impacted by noise while the forward-backward algorithm of CTC loss enforce monotonic alignment between input speech features with target sequences. So the model becomes more robust than purely attention based model. Moreover, using CTC as an auxiliary optimization function speeds up the process of estimating the desired alignment than solely depending on data-driven attention methods.
The right part of Figure 3 illustrates the structure of decoder, which is similar to the encoder, except for the masked multi-head attention module. To prevent attending to future information and preserve the auto-regressive manner in the decoder, the masks in the masked multi-head attention module swept out all values of illegal connections. This masking of the sequence can be achieved in parallel using an elementwise product with a triangular binary matrix. y = (y1, y2, ..., yN) represent the transcriptions of audio data.

\[ L_{\text{CTC}} = \sum_{(k, y)} -\log(P(y|x)) \]  
\[ L_{\text{Attention}} = -\log P(y|x) = -\sum_{u} \log P(y_u^n | x; y_{1:[u-1]}) \]  

where \( y_{1:[u-1]} \) is the ground truth of the previous words. The joint training method of CTC with Transformer works as:

\[ L_{\text{JCT}} = \alpha L_{\text{CTC}} + (1 - \alpha)L_{\text{Attention}} \]

\( \alpha \) is a hyper-parameter: \( 0 \leq \alpha \leq 1 \).

3.3. Fine-tuning methods

We leverage massive unsupervised audio data to train the encoder of JCT. The training process won’t stop until the result in validation dataset triggers the patience of early-stop criteria. Completion of the pre-train process provides high-level representation for down stream tasks. Therefore, we propose two approaches for the fine-tuning stage:

* Directly fine-tuning: Initialize the trainable parameters of encoder in JCT with the results we get from the pre-training process, then use labeled data to optimize the supervised joint loss function (JCT).

* Frozen fine-tuning: Since MPE provides more implicit and high-level representations than Fbank features. In the fine-tuning process, it performs better when we froze the encoder and only trained the parameters of JCT decoder, which means remove the parameters of encoder from the trainable parameters list of JCT. After the accomplishment of decoder training process, for better performance, we can train the whole structure in a supervised manner for a few epochs.

In our experiments, we have explored both two fine-tuning methods, the latter showed much better performance than the former. Essentially, the former fine-tuning method is a simple initialization of encoder in the supervised training stage, integrated with randomly initialized decoder will lose some information we attained from unlabeled data. Thus the difference between directly fine-tuning method and totally supervised training method is very small. While the latter one thoroughly used the representation from massive unsupervised data, it showed much lower word error rate (WER) than totally supervised training in a low resource setting. The result are demonstrated in section 5.

4. Experiments

4.1. Datasets

We made several experiments on LibriSpeech corpus and wall street journal (WSJ) corpus respectively. For LibriSpeech [24] which contains 960 hours training audio data, we used the entire dataset to train MPE for high-level feature extraction. In the fine-tuning process, we exploited train-clean-100 and train-clean-360 for supervised training, dev-clean for validation and text-clean for evaluation. As for WSJ, the models were training on si284 which includes about 81 hours audio data, validating on dev93 and evaluating on eval92. To evaluate the effect of MPE, we leverage the whole dataset for pre-training while one third, a half and the entire data set are respectively used for supervised training. Meanwhile, an ideal feature extractor should extract representations that generalize to datasets of different domains. Thus, to examine the robustness of shifting in domains, we firstly trained MPE on LibriSpeech, then fine-tuned it to JCT with WSJ 81 hours supervised data. We choose totally supervised training on JCT as our baseline.

4.2. Experiment setups

The input acoustic features are 80-dimensional filterbanks extracted with a hop size of 10ms and a window size of 25ms, extended with temporal first and second order differences and per-speaker mean subtraction and variance normalization [3]. The MPE consists of 2 CNN layers with RELU activation function and a stack of 12 encoder blocks. CNN has stride size 2 and kernel size 3 for downsampling. The channels of first layer is 64, next layer has twice as many channels as the previous one. For encoder blocks, each block contains two sub-layers: feed-forward layer (FFL) and self-attention layer (SAL), the dimension of FFL is 2048, as for SAL, the attention heads is 4 and dimension of embedding is 512. The SAL and FFL in the decoder obeys the same configuration, while the number of decoder stacked blocks is set to 6. We used Adam optimizer with default parameter configuration in both two-stage training. Especially in supervised training process, we applied warming up method to vary the learning rate in the whole training process with Noam learning strategy.

\[ lr = k \cdot n^{-0.5} \cdot \min(n^{-0.5}, n \cdot \text{warmup}^{-1.5}) \]

\( k, d, n, \text{warmup} \) respectively refers to a tunable hyper-parameter, model dimension, training step, total warming up steps. The learning rate increased in start warming up \( n \) steps and decreased after the peak of \( lr \). In our experiments, warming up steps \( n = 25000 \), hyper-parameter \( k = 10 \). To avoid over-fitting, label smoothing strategy which was proposed in [20] was also applied in the training process, and the label smoothing weight is set as 0.1. Meanwhile, both of residual dropout and attention dropout [21] were set to 0.1. Moreover, we also used SortaGrad [25] method in the first training epoch for faster convergence and less noise inference. Apart from above configuration, for the multi-task training process, the hyper-paramater \( \alpha \) is set as 0.3.

5. Results

5.1. Pre-training results

![Figure 3: self-attention matrix image of one head in MPE from example4kac031f. The horizontal axis represents input frames to the self-attention block, the vertical axis refers to the output frames of encoder.](image-url)
Table 1: Results on WSJ corpus

| representation | unlabeled | labeled | fine-tuning steps | dev93 | eval92 | baseline(supervised) dev93 | eval92 |
|---------------|-----------|---------|-------------------|------|-------|-------------------------|-------|
| MPE WSJ(81h)  | one-third(25h) | 5500 | 10.43 | 9.31 | 15.05 | 12.54 |
| MPE WSJ(81h)  | half(40h) | 3300 | 7.97 | 7.04 | 12.58 | 10.07 |
| MPE WSJ(81h)  | WSJ(81h) | 15000 | 6.79 | 4.26 | 7.93 | 5.48 |
| MPE LibriSpeech(960h) | WSJ(81h) | 12000 | 8.42 | 4.87 | 7.93 | 5.48 |
| wav2vec[13]   | LibriSpeech(960h) | WSJ(81h) | - | 6.84 | 3.97 | - |
| DeCoAR[16]    | LibriSpeech(960h) | WSJ(81h) | - | 6.30 | 3.17 | - |
| DeCoAR[16]    | WSJ(81h) | WSJ(81h) | - | 8.34 | 4.64 | - |

Table 2: Results on LibriSpeech corpus

| unlabeled data | labeled data | fine-tuning steps | dev clean | test clean | baseline(supervised) dev clean | test clean |
|----------------|-------------|-------------------|----------|-----------|-------------------------------|-----------|
| LibriSpeech(960h) | train-clean-100 | 7500 | 8.12 | 9.68 | 11.63 | 12.17 |
| LibriSpeech(960h) | train-clean-360 | 13000 | 6.44 | 7.83 | 8.35 | 9.70 |
| - | LibriSpeech-960 | - | - | - | 3.24 | 3.77 |

In pre-training stage, in order to measure the reconstruction discrepancy, we have tried L1 loss and huber loss, optimized by Adam optimizer. Although L1 loss has the demerit of slowness convergence, it appeared much better performance after fine-tuning than huber loss, thus we choose L1 loss in pre-training stage. Figure 3 shows the tendency of alignment between original frames and reconstruction frames. From left to right respectively represents the matrix image in epoch1, epoch5 and epoch20. In first epoch, the self-attention matrix image is random but gradually become orthogonal after several training epochs.

5.2. Supervised fine-tuning results

The given results in all these tables are an average of WER in two runs. Specifically, in the decoding stage, we applied beam search (beam width=10), an RNNLM (trained by the transcription of corresponding audio corpus) and CTC decoding method.

Table 3: comparison of two fine-tuning methods

| Fine-tuning methods | unlabeled | labeled | dev93 |
|---------------------|-----------|---------|-------|
| Directly fine-tuning | WSJ(81h) | WSJ(25h) | 14.77 |
| Frozen fine-tuning | WSJ(81h) | WSJ(25h) | 10.43 |

In order to evaluate the two fine-tuning methods we have proposed in section 3.3, a simple experiment on WSJ subset with the two methods has been made. Table 3 revealed freeze encoder method performed far more excellent than simply initialize the encoder. Obviously, directly fine-tuning to supervised model descend the information learned from the pre-training process in follow-up training steps. While freeze the encoder at first performs much better since it avoids the deviation of decoder’s random initialization.

5.2.1. Results on WSJ

The results of WSJ are depicted in Table 1. In WSJ corpus, we select one-third, a half and entire data from it respectively. For comparison, first, we directly trained the three subsets on JCT structure without pre-training. Afterwards, we trained MPE with the whole si284 which contains 81h audio data, then the three subsets were used for supervised training stage. After several trials, compared with directly supervised training, our two stage training achieves 22% wer reduction on dev93 and 30% on eval92. Besides, in order to test the robustness of masked pre-trained method, we applied the MPE which was trained by LibriSpeech-960h and fine-tuned it on WSJ 81h supervised data. We can see from the table that increasing unlabeled data for MPE naturally attains better results.

In bottom half of Table 1 we provide the comparison of MPE and other related published representations: wav2vec, DeCoAR. Wave2vec constructs a five-layer convolutional network. DeCoAR constructs LSTM based netural network. Compared to these two structure, we achieved 15% wer reduction than wav2vec and 7% wer reduction than DeCoAR on dev93. While the result on eval92 behaves not so desirable, we consider that the data set is approaching saturation or we need better RNNLM, we’ll propose several new ideas to address this issue in our future work.

5.2.2. Results on LibriSpeech

Table 2 demonstrates the results of Librispeech subsets. MPE has been trained on 960 hours Librispeech unlabeled audio data, while train-clean-100 and train-clean-360 were chosen to be labeled dataset in fine-tuning stage. The supervised baseline are also given in the table. Compared with the baseline, two-stage training obtained 34% and 25% wer reduction on dev clean and test clean, respectively.

6. Conclusion

According to all the above experiments, two-stage training has significantly remarkable performance better than all-supervised training. It suggested that with massive unlabeled data and limited labeled data we can achieve the same performance with the system which has been trained by a large amount of supervised data. Meanwhile, relying on the powerful modeling ability of Transformer, the masked pre-trained representation can be widely used to other down stream speech tasks.
7. References

[1] Barlow, H. B. “Unsupervised learning”. Neural computation, (1989): 295-311.

[2] Devlin J, Chang M W, Lee K, et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. arXiv preprint arXiv:1810.04805, 2018

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need CoRR, vol. abs/1706.03762, 2017

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Recurrent temporal classification: labelling unsegmented sequence data with recurrent neural networks in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 369-376.

[5] D. Linhao, X. Shuang, and X. Bo. Speech-transformer: a non-recurrence sequence-to-sequence model for speech recognition in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5884-5888.

[6] Z. Shiyu, D. Linhao, X. Shuang, and X. Bo, Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese. arXiv preprint arXiv:1804.10752 2018

[7] Z. Shiyu, D. Linhao, X. Shuang, and X. Bo, A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese. arXiv preprint arXiv:1804.10752 2018.

[8] Z. Shiyu, X. Shuang, and X. Bo. Multilingual-end-to-end speech recognition with a single transformer on low-resource languages. arXiv preprint arXiv:1806.05059 2018

[9] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney A comparison of transformer and lstm encoder decoder models for asr in IEEE Automatic Speech Recognition and Understanding Workshop, Sentosa, Singapore, 2019.

[10] A. Graves. Sequence transduction with recurrent neural networks CoRR, vol. abs/1211.3711, 2012

[11] H. Sak, M. Shannon, K. Rao and F. Beaufays Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping in Proc. Interspeech 2017, 2017, pp. 1298-1302.

[12] Muda, Lindasalwa, M. Begam, I. Elamvazuthi. “Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques”. arXiv preprint arXiv:1007.4083 2010

[13] Y. Ye and X. Guangxu and Suo. Quiling and J. Kebin and Zhang. Aidong. “Wave2vec: Learning deep representations for biosignals” in 2017 IEEE International Conference on Data Mining (ICDM)IEEE, 2017.

[14] Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding.” arXiv preprint arXiv:1807.03748 2018.

[15] C. Yu-An, H. Wei-Ning, T. Hao , and J. Glass. An unsupervised autoregressive model for speech representation learning, Interspeech, Sep 2019.

[16] L. Shao-shi and L. Yu-zong and S. Ju-lian and K. Katrin. “Deep Contextualized Acoustic Representations For Semi-Supervised Speech Recognition”. arXiv preprint arXiv:1912.01679 2019

[17] L. Andy T and Y. Shu-wen and C. Po-Han and H. Po-chun and L. Hung-yi. “Mockingjay: Unsupervised Speech Representation Learning with Deep Bidirectional Transformer Encoders” arXiv preprint arXiv:1910.12658 2019

[18] A. Mohamed, D. Okhonko, L. Zettlemoyer. Transformers with convolutional context for asr, arXiv preprint arXiv:1904.11660 2019.

[19] S. Kim, T. Hori, and S. Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning in 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017, pp. 4835-4839.

[20] R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? CoRR, vol. abs/1906.02629, 2019.

[21] Baldi, P. Sadowski, P. J. “Understanding dropout.” In Advances in neural information processing systems (pp. 2814-2822), 2013

[22] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen et al. Deep speech 2: End-to-end speech recognition in english and mandarin, in International conference on machine learning, pp. 1731-182, 2016

[23] X. Aubert, C. Dugast, H. Ney and V. Steinbiss, “Large vocabulary continuous speech recognition of Wall Street Journal data,” Proceedings of ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing, Adelaide, 1994

[24] Panayotov, Vasil, et al. “Librispeech: an asr corpus based on public domain audio books.” 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015.