
Semi-supervised learning with soften probabilistic labels

Chun Hung Lin, Heng Fang, Saga Abdul Amir, Weihong Sung
KTH Royal Institute of Technology
chlin3@kth.se, hfang@kth.se, sagaaa@kth.se, weihongs@kth.se,

Abstract

We studied a novel idea to combine the knowledge distillation concept into semi-supervised learning method for acoustic modelling with limited amount of annotated training data. Our method consists of a pair of teacher and student model in which the teacher model trained with labeled data creates probabilistic labels at temperature T and the student is then trained with those probabilistic labels. In this study, we trained 3-layer bi-directional LSTM as the teacher model and 3-layer LSTM as the student model as well as a baseline model identical to the student model and evaluated them. Our experiments show that the teacher-student method increases the accuracy with the help of data without annotation.

1 Introduction

The goal of automatic speech recognition is to provide a mapping from sequences of acoustic frames to sequences of linguistic symbols (phonemes or words). Recently, with the development of artificial intelligence and machine learning, various deep models have been used to complete the task of speech recognition, such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Long Short Term Memory network (LSTM). Although these methods perform pretty well in the field of speech recognition, they are supervised learning algorithms which require the datasets all labeled by the machine learning engineer or data scientist. The workload of labeling a large dataset is extremely time-consuming, hence several unsupervised learning algorithms are proposed.

On the contrary, unsupervised learning algorithms are trained on unlabeled data and they would automatically determine the structure in the data by extracting useful features and analyzing the relationship between the features and the desired outcomes. K-Means clustering, Gaussian Mixture Models (GMM), Hidden Markov Models (HMM) and Principal Component Analysis (PCA) are all examples of unsupervised learning. Nevertheless, there are also some problems with unsupervised algorithms. Since the data they trained don't have ground truth labels, it is difficult to measure the accuracy of the algorithms based on unsupervised learning.

Inspired by the strengths and weaknesses of both supervised and unsupervised learning, Blum and Mitchell proposed a new trade-off concept called co-training in [1], and co-training gradually evolved into today's concept of "semi-supervised learning".

Semi-supervised learning is a technique that deals with problems involve a lot of unlabeled data and a huge narrow labeled data. Labeling all the data in reality is unrealistic and expensive, but the deep learning network can still benefit from the small proportion of labeled data and achieve a better performance than totally unsupervised learning.

In this project, we would like to implement a semi-supervised learning system and compare it with a supervised learning system.

Inspired by Hinton's knowledge distillation (student/teacher training)[4], we trained a teacher Bi-directional LSTM model on a small set of labeled data and use it to create probabilistic labels on a

larger set of unlabeled data. We then trained a student LSTM model with probabilistic labels of the unlabeled data generated by the teacher model. This teacher and student pair combined as the semi-supervised learning model.

For the supervised learning model, we trained a LSTM model of which the network architecture is completely identical to the student architecture as the baseline model for comparing with the semi-supervised learning model.

The report is organised as follows: Section 2 introduces several related works which inspire our project. Section 3 describes the methods. Section 4 records details on the experimental setup. Section 5 reports the corresponding experimental results. Section 6 proposes discussions of the experiments and concludes the report.

2 Related Work

Semi-supervised learning has a long history in the field of acoustic speech recognition. In [7] Kemp and Waibel proposed a semi-supervised algorithm which trained a speech recognizer with only a minimal amount (30 minutes) of transcriptions and a large amount (50 hours) of untranscribed data. They employed the bootstrap recognizer to generate initial hypothesis and successfully improved the accuracy by 10.5%. In [9] Lamel, Gauvain and Adda used their experiments to demonstrate that lightly or unsupervised learning can greatly reduce the cost of building acoustic speech recognition models. In [12] Ma, Matsoukas, Kimball and Schwartz successfully applied lightly and unsupervised learning algorithms to English and Arabic Broadcast News recognition system.

Recently, due to the popularity of deep learning, there are also a lot of research work combining semi-supervised learning and deep neural networks (DNN). In [6] Huang, Yu, Gong and Liu proposed and implemented semi-supervised GMM-HMM model and DNN-HMM acoustic model on the short message dictation task, and achieved 85%- 92.8% high accuracy. Meanwhile, in [17] Thomas, Seltzer, Church and Hermansky proposed a new approach called deep neural network features and semi-supervised learning to build the front-end features and managed to apply it to large vocabulary continuous speech recognition in low resource settings.

Our project is mainly inspired by [2] and [13]. In [2] Dhaka and Salvi proposed a semi-supervised learning method which combined sparse auto-encoders with feed-forward networks to improve the acoustic speech recognition system, and their experiments demonstrated that feeding the neural networks with additional unlabeled data would boost the system's performance. In [13] parthasarathi and Strom from Amazon introduced the concept of knowledge distillation and student-teacher training on the unlabeled data to acoustic models, which would help to expand target generation.

3 Method

3.1 Long Short Term Memory (LSTM)

LSTMs have been implemented to advance the research in many fields, such as speech synthesis, language modeling and translation, handwriting recognition and protein structure prediction. They were introduced by Hochreiter and Schmidhuber [5] in 1997 who designed it to handle the problem of long term dependencies. There exist many variants of this network today. In general, LSTMs consist of a chain of repeating sections of the neural network, and they are called loops (unraveled). LSTMs are similar to RNN, but there are still quite some differences. In the traditional RNN, the structure of the repeating section is simple and may consist only one neural network layer, where LSTMs allow more complex structures. We can have more hidden layers in different ways.

Each line we see in Figure 1 is a vector, from the output of one node to the inputs of others. The pink circles we see above are point-wise operations and the yellow boxes are the learned neural network layers.

Understanding the LSTM architecture is crucial. The first step in a LSTM is to decide which information to throw away and which to keep, and this can be done by a sigmoid layer that we call the "forget gate" layer.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

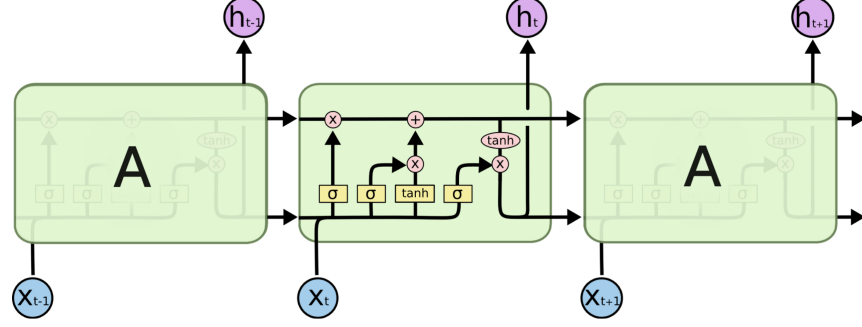


Figure 1: Long Short Term Memory LSTM from [16]

This layer uses h_{t-1} from the previous cell state and a new input value x_i as inputs, and outputs a number between 0 and 1 where 0 means forget completely and 1 means remember all. The second step is to employ another sigmoid function called the "input gate" to decide the information we should keep, which decide the value we should update and where to store this value.

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

The third step is the "tanh" layer which we called for the candidate gate layer. In this layer we scales the input and create a vector of the new candidates. We multiply the candidates, then push them to the old cell state and let them multiply with the forget gate layer to get a new value.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c[h_{t-1}, x_t] + b_f)$$

The last step is to decide which output we should use for the next cell state. Each layer in the cell states have their own weight-matrices and biases where they are updated iteratively with the back propagation using the gradient of cross entropy loss against the true target.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

3.2 Bidirectional LSTM (Bi-LSTM)

As we can see from Figure 1, the architecture of LSTM is forward, and it only takes advantage of past information. However, since we have the complete utterance, it is also possible to take advantage of future information[10]. In consideration of that, a new architecture called Bidirectional LSTM (Bi-LSTM) was proposed. Figure 2 shows the structure of Bi-LSTM. Bi-LSTM has two opposite directions - forward and backward. The forward direction unfolds the networks from the front to the end just like a normal LSTM, while the backward direction does the reverse from the end to the front. The two directions of Bi-LSTM work in parallel with their unique parameters (unique weights and biases). Hidden states in both directions are concatenated together and sent to the higher layer. In the end of forward direction LSTM, hidden states are converted into state probabilities using a softmax classifiers.

3.3 Student-Teacher Learning

In order to create labels for unlabeled data, a teacher network is trained with a small amount of labeled data and create a set of probabilistic labels for those unlabeled data. [13] In other contents, the probabilistic labels also called soft targets.

Ideally, the teacher model is expected to be more powerful and complex than the student model. Therefore, we picked up this idea in the experiment.

With inspiration from the work of Hinton et al. [4], we designed the probabilistic labels for the class i as

$$p_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \quad (1)$$

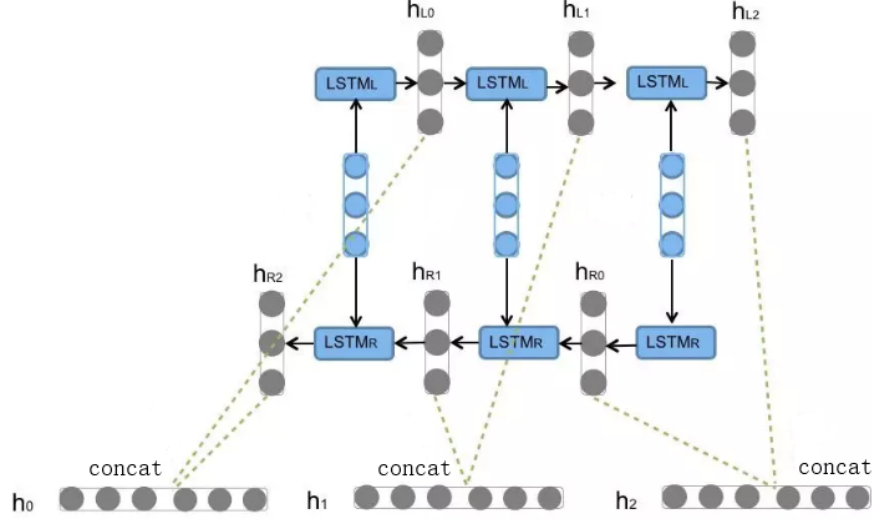


Figure 2: Bi-directional LSTM

where z_i is logit output which are usually converted to a class probability p_i by a softmax layer, and T is a hyperparameter called temperature.

The temperature is to soften the probabilistic labels in order to transfer more knowledge and information via the probabilistic labels. If the temperature equals to 1, the probabilistic labels will reduce to the softmax layer outputs. The softmax layer filters out relatively unlikely classes and keep only one or two the most likely classes. If the temperature is higher, labels calculated from eq. (1) are softer [4] and the information like top-k classe probabilities can be transferred with the probabilistic labels.

To train the student network with unlabeled data, we minimize the cross-entropy between the student and the teacher probabilistic labels at the same temperature. As the cross entropy has the following relation with Kullback–Leibler divergence:

$$D_{KL}(p^{\text{teacher}} || p^{\text{student}}) = H(p^{\text{teacher}}, p^{\text{student}}) - S(p^{\text{teacher}}) \quad (2)$$

where $H(p^{\text{teacher}}, p^{\text{student}})$ is the cross entropy of between the class probability distributions predicted by the teacher model and the student model and $S(p^{\text{teacher}})$ is the entropy of the class probability distribution by the teacher. Since we trained the student model with a pretrained teacher model and the class probability distribution from the teacher did not change. Therefore, the entropy term $S(p^{\text{teacher}})$ is constant during the student training phase and minimizing $D_{KL}(p^{\text{teacher}} || p^{\text{student}})$ is equivalent to minimizing $H(p^{\text{teacher}}, p^{\text{student}})$. As the result, we have the objective function for student network as:

$$L = T^2 D_{KL}(p^{\text{teacher}} || p^{\text{student}}) \quad (3)$$

$$p^{\text{teacher}} = \frac{\exp(z_i^{\text{teacher}}/T)}{\sum_j \exp(z_j^{\text{teacher}}/T)} \quad (4)$$

$$p^{\text{student}} = \frac{\exp(z_i^{\text{student}}/T)}{\sum_j \exp(z_j^{\text{student}}/T)} \quad (5)$$

where z_i^{teacher} and z_i^{student} are the class logits before the softmax layer and p_i^{teacher} and p_i^{student} are the predicted class probabilities at temperature T from the teacher and student respectively. The temperature T is added to the cost function in order to have a comparable order of magnitude across different temperatures [4].

4 Experiments

4.1 Dataset

In this study we used the standard TIMIT dataset [3] to investigate the frame-based phoneme classification for our designed method. We used 3521 sentences as the training set, 184 sentences as the validation set and 192 sentences as the test set from the standard core test material. In order to compare the results in other literature, glottal stop segments are excluded. The feature vector has 39 dimensions, which consist of 12 MFCC coefficients, 1 energy coefficient, delta, and delta-delta. Delta, and delta-deltas are the first order and the second order time derivatives of the MFCC coefficients. The total number of frames for the training, validation, and test sets are 1057430, 52893, and 56926 respectively.

During the training phase, we used all 48 phones as target classes while they collapsed into 39 phones during evaluation.

To emulate unlabeled data for semi-supervised learning, we drew 1%, 3%, 5%, 10%, 20%, 30%, and 50% of training set as the labeled data and the rest of training set became unlabeled data. The teacher and the baseline model were trained only with labeled data. The teacher-student model was trained with the whole training set but the labels are generated by the teacher model and the original labels were ignored. As we have multiple teacher models trained from different amounts of labeled data, we have multiple teacher-student models.

4.2 Networks and their architectures

For the network architectures, we used 3-layer Bi-LSTM and 3-layer LSTM as the teacher model and the student model respectively. Since the teacher model is expected to be more powerful than the student model, we chose the Bi-LSTM as the teacher and the LSTM as the student. The baseline model is identical to the student model except the different types of training data.

We used one fully connected layer as the output layer which is for transforming hidden layer nodes to target classes. The dimension of input features was 39 and the number of output classes was 48. The number of hidden nodes was 96 which is the twice of the output classes.

4.3 Training details

It is well-known that the learning rate is a crucial hyper-parameter to train a deep network. [15] We first did a small preliminary comparison on which learning rate scheme we should employ. We compared three schemes and they were Cyclical Learning Rates (CyclicLR in short), Adam [8] and decaying the learning rate by 10 each 10 epochs (StepLR in short). We set the number of training epochs to be 50. The maximum and the minimum of the learning rate for cyclical learning rate were 10^{-2} and 10^{-5} . The step size for a half cycle was the number of iteration for two epochs. For Adam, we set the betas to be 0.9 and 0.999 for computing running averages of gradient and its square. For Adam and StepLR, the initial learning rate for both methods were 10^{-2} . We used mini-batch gradient descent and the batch size were 100. We did not use momentum in mini-batch gradient descent and we set the weight decay (L2 penalty term weighing) to be 5×10^{-4} . We used a padding technique to pack variable length of sequences into a batch and used a masking technique to perform the back-propagation with the automatic differentiation provided by pytorch. [14]

We used 30% of the training data and the same set of validation as well as the test data to compare the learning rate schemes. The comparison results are shown in fig. 3. The results show that the CyclicLR scheme performed better in both validation accuracy and test accuracy and therefore we used CyclicLR to do the whole experiment.

Therefore for the experiment, we used CyclicLR with the same setting in the preliminary comparison test.

4.4 Temperature Searching

In the semi-supervised learning model we designed, we the temperature T as a hyperparameter and we optimized it with the validation set. We performed a grid search for the best temperature for

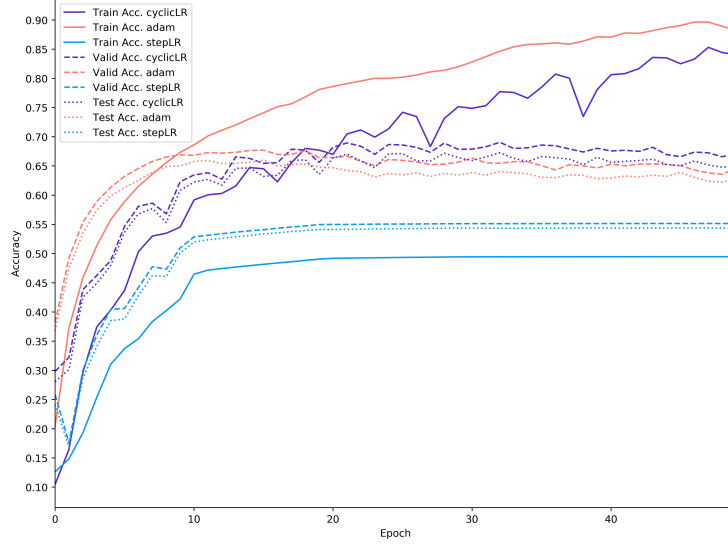


Figure 3: Preliminary comparison for different learning rate schemes

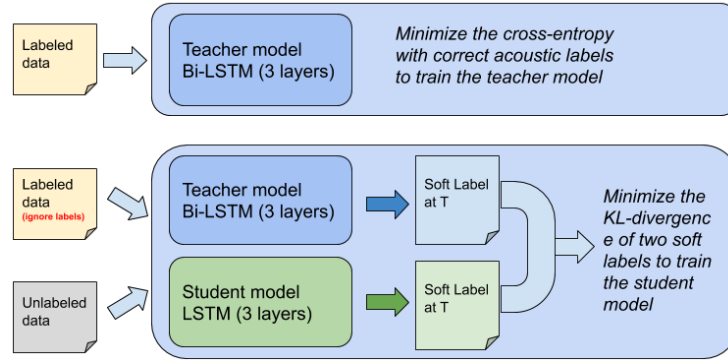


Figure 4: The graphical representation of how to train the teacher and teacher-student models

different proportions of labeled data in training set. The searched temperatures were 0.5, 1, 2, 4, 6, 8, 10, 20, 50, 100, 200, 500, 800.

5 Results

In this section we will mainly post the results of our experiment and the detailed analysis will be discussed in the next section.

The frame-based phoneme classification performances against the percentages of labeled data are shown in table 2. Comparing our teacher-student model with the baseline model, which is the student model trained with labeled data only, the unlabeled data gives more extra information from the teacher to the student model and therefore the performance of the semi-supervised learning is much better than the supervised learning baseline model. fig. 5b shows the same result.

The validation accuracy performed in grid searching of the temperature parameter are shown in fig. 5a. We can see the difference in accuracy under the same percentage of labeled data is in 2%.

Moreover, more labeled data increase both supervised learning model and semi-supervised learning method. This result is expected as more information brought by professional human annotators.

Our LSTM or bi-LSTM method is comparable to the state-of-the-art graph-based semi-supervised learning techniques as shown in table 1.

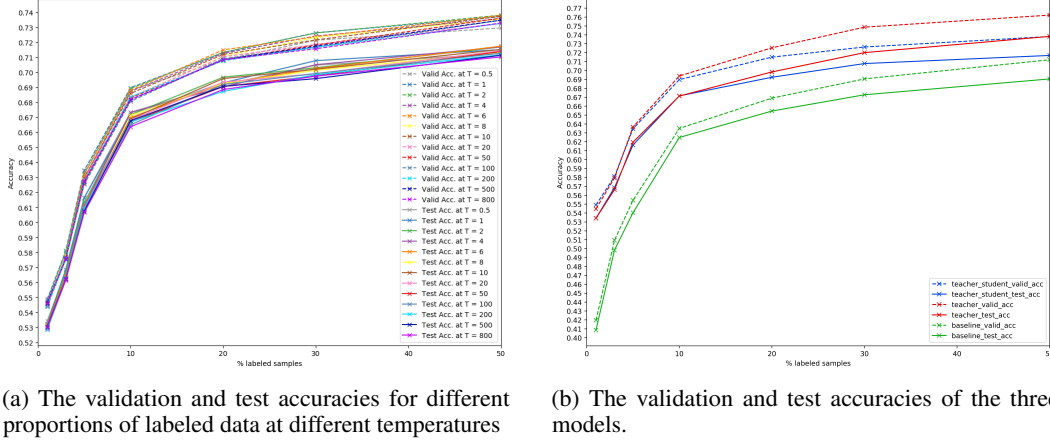


Figure 5: For fig. b, the teacher-student models are at the optimal temperature on validation set. The corresponding values can be found in table 2.

Table 1: Frame-based phoneme recognition accuracy against percentage of labelled training examples on the TIMIT dataset. The first 6 networks were trained in supervised way and the last two were trained in semi-supervised way.

Comparison with other methods				
Method	Reference	10% labelled	30% labelled	
		Test accuracy(%)		
NN	[2]	65.94	69.24	
LP	[11]	65.47	69.24	
MP	[11]	65.48	69.24	
MAD	[11]	66.63	70.25	
pMP	[11]	67.22	71.06	
LSTM (3 layers)	this work	62.48	67.28	
SSSAE	[2]	67.03	69.65	
Teacher-student	this work	67.15	70.78	

Table 2: Results on frame-based phoneme classification on the test and validation sets on the TIMIT from the three models.

Labelled Obs.		3-layer LSTM (baseline)		Teacher-Student			3-layer Bi-LSTM (teacher)	
%	#	valid. acc.(%)	test acc.(%)	valid. acc.(%)	test acc.(%)	Opt. Temp	valid. acc.(%)	test acc.(%)
1	10704	41.96	40.86	54.90	53.42	0.5	54.47	53.42
3	31797	50.96	49.84	58.10	56.86	1	57.94	56.63
5	53139	55.43	54.04	63.46	61.64	1	63.67	61.95
10	107548	63.50	62.48	68.98	67.15	1	69.39	67.14
20	212399	66.90	65.45	71.50	69.24	6	72.55	69.84
30	314517	69.06	67.28	72.64	70.78	1	74.87	72.01
50	523193	71.21	69.06	73.82	71.69	2	76.22	73.83
100	1057430	73.98	72.58	74.86	73.24	1	77.87	76.50

6 Discussion and Conclusions

We reported results on frame based phoneme classification on the TIMIT database using semi-supervised learning based on our novel approach. We also made the comparison between our semi-supervised learning method with the supervised learning method. Although our teacher-student model doesn't perform the best, it still outperforms most of the other models including the 3-layer LSTM baseline model. That is to say, our methods outperforms the the 3-layer simple LSTM model trained with back propagation through time algorithm on the same amount of labelled data. Therefore, our student-teacher learning method is meaningful and it improves the test accuracy by 3% ~ 8% compared to the simple baseline model.

Besides, we also did three other experiments. Firstly, we compared the different learning rate schemes as fig. 3 showed, and CyclicLR outperformed in both validation and test accuracy though Adam performs better in training accuracy, so we chose CyclicLR because it was more robust than stepLR and Adam. Secondly, we compared the performance of teacher-student model, teacher model and baseline model on validation and test set. The result was shown in fig. 5b. As previously stated, the teacher-student model always outperformed the baseline model. Moreover, if the percentage of the labeled data was low, there was almost no difference between teacher and teacher-student's performance, that is to say, student can obtain all the information from teacher learned. However, when this percentage became larger, the gap between their performances was also increasing due to the model competence. In this case, teacher model is more complex, and it has greater generalization ability than the student model. In addition, we found that using 20% or 30% of labeled data gave a comparable performance as fully labeled data in teacher-student model. We think it is because the capability of the student model as the baseline model trained with fully labeled data has the test accuracy of 72.58 % while the teacher-student model has the test accuracy of 73.24 %. Finally, we also measured the effect of changing different T values. fig. 5a shows that although teacher-student model performed similar behavior under different T values, it still affected the performance, so we should choose its value according to the validation accuracy.

The current state-of-art phoneme recognition performance on TIMIT dataset is 83.5% accuracy [18], and it performs much better than our methods. The reason is that we only use a part of labelled data for training while they use the whole labelled data as the training set. Moreover, our purpose is to investigate and compare the performance of the semi-supervised learning and the fully supervised learning, so we don't use the mel-scaled time-frequency representation as input features which is used in [18]. Instead, we used MFCCs features in order to do a fair comparison with [2] and [11].

For further implementation, we can mainly do two improvements. The first is that we can use hard label in teacher-student training when calculating the loss. In this project, we only consider the soft label and try to minimize the KL loss of teacher's and student's soft label, but in the future, we can consider the hard label of our labeled data when calculating the loss term. That is, we will have another hyper-parameter λ to control the balance the soft target loss and the hard target loss. The other improvement is we can compare different combination of teacher-student models. We just used Bi-LSTM and LSTM as teacher and student models in our experiments because we focused on exploring whether there will be an improvement on performance if we use the teacher model to train a student model. Therefore, since we have known teacher-student model is useful, we can try different combination of models to achieve the better performance on the test set.

References

- [1] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [2] Akash Kumar Dhaka and Giampiero Salvi. Sparse autoencoder based semi-supervised learning for phone classification with limited annotations. In *Proc. GLU 2017 International Workshop on Grounding Language Understanding*, pages 22–26, 2017.
- [3] William M Fisher. The darpa speech recognition research database: specifications and status. In *Proc. DARPA Workshop on Speech Recognition, Feb. 1986*, pages 93–99, 1986.
- [4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.
- [6] Yan Huang, Dong Yu, Yifan Gong, and Chaojun Liu. Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration. In *Interspeech*, pages 2360–2364, 2013.
- [7] Thomas Kemp and Alex Waibel. Unsupervised training of a speech recognizer: Recent experiments. In *Sixth European Conference on Speech Communication and Technology*, 1999.
- [8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [9] Lori Lamel, Jean-Luc Gauvain, and Gilles Adda. Lightly supervised and unsupervised acoustic model training. *Computer Speech & Language*, 16(1):115–129, 2002.
- [10] X Liu. Deep convolutional and lstm neural networks for acoustic modelling in automatic speech recognition, 2018.
- [11] Yuzong Liu and Katrin Kirchhoff. Graph-based semi-supervised learning for phone and segment classification. In *INTERSPEECH*, pages 1840–1843, 2013.
- [12] Jeff Ma, Spyros Matsoukas, Owen Kimball, and Richard Schwartz. Unsupervised training on large amounts of broadcast news data. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 3, pages III–III. IEEE, 2006.
- [13] Sree Hari Krishnan Parthasarathi and Nikko Strom. Lessons from building acoustic models with a million hours of speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6670–6674. IEEE, 2019.
- [14] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [15] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.
- [16] SuperDataScience. Recurrent neural network (rnn) - long short term memory (lstm), 2018.
- [17] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky. Deep neural network features and semi-supervised training for low resource speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6704–6708. IEEE, 2013.
- [18] László Tóth. Phone recognition with hierarchical convolutional deep maxout networks. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):25, 2015.

Appendix

The followings are provided by the reviewers. Under each suggestion, we stated if we adopted the suggestion or we rejected the suggestion with explanation.

Reviewer 1

1. Adding more papers in addition to X Liu.

We have carefully read the three papers which was suggested by reviewer 1, and found that the first two papers emphasized bidirectional LSTM and the third one covered more insight in TIMIT dataset. However, we think that we don't need to add these papers, since our project is more focused on student-teacher training and bidirectional LSTM is just a baseline for us to do comparison. Besides, our reference [2] already contains the introduction of TIMIT dataset.

2. Unclear about cross-entropy and the temperature

In the student-teacher training, there are two kinds of cross entropy. The first one is the cross-entropy with the soft targets which are computed under the same temperature, and the second one is the cross-entropy with the correct labels, or you can say that the correct labels is the circumstance when the temperature is 1. For more details, please refer to our reference [4].

3. summarize the all training process and more training details

See our modified section 4.3 Training details.

4. More details on the equations of teacher-student training

See our modified section 3.3 Student-Teacher Learning.

5. Add LSTM equations

Our project target is the semi-supervised learning and the teacher-student training method. We just mention the brief idea of LSTM and why it can be used in acoustic model. Readers are expected to learn LSTM themselves otherwise the report will be too verbose and lengthy.

Reviewer2

1. Reference the plot of LSTM structure

Done. We have added reference [16].

2. Reference TIMIT database

Done. We have added reference [3].

3. Reference the optimization scheme

Done. Added reference for adam and cyclic learning rate scheme

4. Details of the LSTM network as well as Bi-LSTM network

The details are introduced in section 3.1 and 3.2. During our experiments, we used 3-layer LSTM and Bi-LSTM with 96 hidden nodes(twice of the output class number).

5. Why use adam?

Using Adam has several advantages: efficient calculation, good at solving sparse gradient or high noise problem, etc.. However using Adam usually converges to local minima so it is not so robust as using cyclic LR.

Reviewer3

1. Suggest to use TIMIT reference.

Duplicated. The feedback resolved.

2. More discussion of temperature.

The temperature is for softening the soft targets. Added brief explanation how temperature affect the targets.

3. Elaborate LSTM method and theory.

Same answer as the review 1 question 5.

4. What is the meaning of phoneme and would you explain it?

We think that reviewers are competent and did the laboratory well.

5. Discussion on why there are studies that give way better accuracies on the same task.

Our target is to show the semi-supervised learning is feasible under our purposed method. Our target is not to perform better than the state-of-the-art method. In our experiment, we proved that the teacher-student model works and better than the baseline model.

Reviewer 4

1. How the labels you use in the semi-supervised are chosen among the available ones?

See section 4.1 for details

2. Is these uniform in the classes?

The purposed acoustic model is to classify the extracted features vector to a phoneme which is what the class the reviewer means. We did not have a distribution of phoneme on hand.

3. State clear on what is the goal and rationale of each approach you used.

Stated in introduction. We did a small test for choosing the training scheme.

4. State more on why you make the decision.

Our focus point is the semi-supervised learning and we purposed the idea with explanation how the idea comes from. We think it is more than enough.