Variational Connectionist Temporal Classification

Linlin Chao, Jingdong Chen, Wei Chu

Ant Financial Services Group
{chulin.cll, jingdongchen.cjd}@antgroup.com
weichu.cw@alibaba-inc.com

Abstract. Connectionist Temporal Classification (CTC) is a training criterion designed for sequence labelling problems where the alignment between the inputs and the target labels is unknown. One of the key steps is to add a blank symbol to the target vocabulary. However, CTC tends to output spiky distributions since it prefers to output blank symbol most of the time. These spiky distributions show inferior alignments and the non-blank symbols are not learned sufficiently. To remedy this, we propose variational CTC (Var-CTC) to enhance the learning of non-blank symbols. The proposed Var-CTC converts the output distribution of vanilla CTC with hierarchy distribution. It first learns the approximated posterior distribution of blank to determine whether to output a specific non-blank symbol or not. Then it learns the alignment between non-blank symbols and input sequence. Experiments on scene text recognition and offline handwritten text recognition show Var-CTC achieves better alignments. Besides, with the enhanced learning of non-blank symbols, the confidence scores of model outputs are more discriminative. Compared with the vanilla CTC, the proposed Var-CTC can improve the recall performance by a large margin when the models maintain the same level of precision.

Keywords: Connectionist Temporal Classification, Scene text recognition, Handwritten text recognition

1 Introduction

Connectionist Temporal Classification (CTC) [4] is a training criterion designed for sequence labelling problems where the alignments between the inputs and the target labels are unknown. It has gained widespread traction from its successful use in tasks such as speech recognition [5, 7, 24], text recognition [6, 30], sign language recognition [2], video segmentation [13] and so on. It is proven to be effective in sequence recognition tasks.

CTC works by adding an extra blank symbol to target vocabulary and maximizing the probabilities of all possible alignments. The added blank symbol represents outputting either a specific non-blank symbol or not. With the added blank symbol, the outputs over all timesteps are aligned to multiple paths, which consists of labels and blanks. The CTC-based training is then to sum up probabilities of all the corresponding paths and maximize them. However, the distribution of blank and non-blank symbols in the training data is unbalanced. This is because: 1) blank is almost added into every training data to make paths; 2) compared with the non-blank, the positions of blanks in paths are more flexible, which leads to more blanks are added. The unbalanced distribution leads to the model prefers output blank most of the time, which is known as the CTC spiky distribution problem [4, 24, 28]. As shown in Fig. 1, the outputs of the characters in label sequence only exist in only a few timesteps. The spiky distributions show inferior alignments [28]. The learning of non-blank symbols is not sufficient, which is suppressed by the added blank.

In this paper, we try to enhance the learning of non-blank symbols by proposing the variational CTC (Var-CTC). The Var-CTC approximates the posterior distribution of blank using a learned inference network. It is fit with variational inference [16] technique to improve the training bound. In the proposed Var-CTC, the influence of the unbalanced distribution for non-blank symbols is relieved. This is because the distributions of blank and non-blank symbols are not belonging to the same one. Var-CTC first learns the approximated posterior distribution of blank to determine whether to output a label or not. Then it learns to determine to output which label. This hierarchy output of Var-CTC is similar to the classification branch in objection detection [26], where the first hierarchy determines the background and non-background category and the second hierarchy determines the object category.

Besides, we find the confidence scores of the CTC model's predictions in text recognition are not discriminative enough. Confidence scores are important for practical use, as we want the model's predictions can achieve the desired precision and recall at the same time. Having a reliable confidence score is crucial for real world applications of OCR. As far as we know, the confidence score based evaluation has never been compared. In this paper, we add the confidence score based evaluation for model comparisons. We find the proposed Var-CTC can improve the Precision-Recall performance significantly.

The main contributions of this paper can be summarized as follows: (1) Variational inference for CTC is first introduced, which converts output distribution of vanilla CTC with hierarchy distribution; (2) The confidence scores of CTC based models on text recognition are analyzed. We show why the confidence score is not discriminative enough by case study and Precision-Recall curve; (3) With the enhanced learning of non-blank symbols, the proposed Var-CTC can improve the recall by a large margin while maintain the same level precision.

2 Related Work

CTC has been explored extensively in sequence recognition tasks, like text recognition [6, 30], speech recognition [4, 5, 7] and so on. A relevant and recent work to ours is the EnEs-CTC [20]. EnEs-CTC proposes a maximum conditional entropy based regularization method to penalize spiky distributions. The penalization

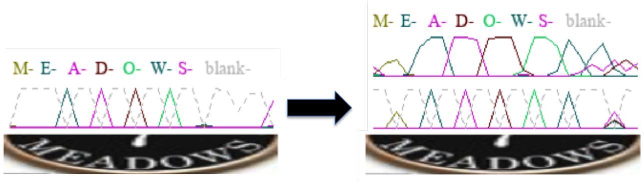

Fig. 1: (Better viewed in color). Visualization of the output distributions for CTC (left) and Var-CTC (right). For Var-CTC, we visualize two distributions, where the bottom one denotes $P(classes|blank, img) * P(blank|img)$ and the top one denotes $P(classes|img)$. The classes in this case represents the characters in label sequence "MEADOWS".

operation enables the model to explore more paths for the sequence alignments, which improves the learning of the non-blank symbols. In contrast to them, our work improves the learning of non-blank symbols by changing the output distribution to hierarchy distribution. The hierarchy distribution relieves the influence of the unbalance problem, which is more thoroughly.

For large scale speech recognition, [32] introduces and evaluates Sampled CTC to speed up CTC training. Two sampling methods are proposed to sample the blank outputs, which are heuristic. Once the blank is sampled, the other parameters can be optimized by minimizing cross entropy objective. Our proposed Var-CTC can also sample the blank output. While the sampling procedure is guided by the approximated posterior distribution, which is end-to-end learnable. GTC [12] tries to learn better alignment with an attention decoder as a guidance for CTC training while this paper focuses on the underfitting for non-blanks in vanilla CTC.

Besides, [3] modifies the CTC by fusing focal loss with it and thus makes the model to attend to the low-frequent samples at training stage. The proposed method tries to solve the class unbalance problem of Chinese optical character recognition. Compared to them, our work try to solve the unbalance problem between blank and non-blank symbols rather then unbalance among non-blank symbols. For weakly-supervised action labelling in video, [13] introduces the extended CTC framework to enforce the consistency of all possible paths with frame to frame visual similarities. However, computing the visual similarities between consecutive frames is expensive.

Meanwhile, CTC based model for scene text recognition has achieved relative high recognition accuracy on several benchmark datasets. These models are often evaluated by sequence accuracy [30, 20]. The confidence score is important but never analyzed. In this paper, we add the confidence score based performance as an evaluation criterion.

2.1 Methodology

Before proceeding, we define our mathematical notations. The input feature sequence is represented as $X = \{x^1, x^2, x^3, ..., x^T\}$, where T is the sequence

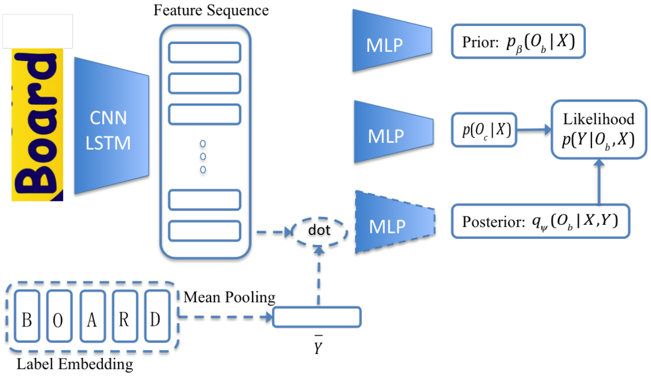

Fig. 2: The probabilistic graphical model of our proposed method. Dotted lines represent the calculation of the approximate posterior distribution. X and Y represent feature sequence and label sequence respectively. O_b represents whether to output a label or not at each timestep. $p(O_c|X)$ represents the categorical distribution for elements in label sequence Y, which does not contain blank.

length, and x^i is a vector representation in $\mathbb{R}^d$. The target label sequence is represented as $Y = \{y^1, y^2, ..., y^{T^y}\}$. T^y represents the length of target label sequence, which is no greater than T. The elements of label sequence come from the target vocabulary A. The blank symbol is represented as "$-$". The extended vocabulary $A \cup \{-\}$ is represented as A^*. The model output is represented as O and it contains blank output sequence O_b and non-blank output sequence O_c. o^t represents the output at timestep t.

2.2 Connectionist Temporal Classification

Given feature sequence X and label sequence Y, CTC learns the alignment without employing the frame level alignment information. The blank symbol "$-$" is added to the target label vocabulary for two reasons. Firstly, it separates the repeated label in the label sequence. Secondly, it is used as label for unlabeled data. At every timestep, the softmax function normalizes the outputs to get the distribution from x^t to A^*.

The complete sequence of outputs is then used to define a distribution over all possible alignments, where each possible alignment is named as a path π. The path π is composed by labels in Y and blanks. Assuming the outputs at each timestep to be independent of those at other timesteps, the probability of one particular path π can be calculated as:

$$p(\pi|X) = \prod_{t=1}^{T} p(o^t_{\pi^t}|x^t). \tag{1}$$

where π^t represents the label at the timestep t for path π. CTC then defines a many-to-one mapping function F. The mapping function F maps the paths to the label sequence by first merging the consecutive same labels into one and then discards the blanks. For example $F(a, -, a, b, -) = F(a, a, -, a, b) = aab$. The probability of label Y can be calculated as an aggregation of the probabilities of all possible CTC paths:

$$p(Y|X) = \sum_{\pi \in F^{-1}(Y)} p(\pi|X). \tag{2}$$

For CTC based models, the CTC is usually applied on the top of bidirectional recurrent neural networks (RNNs)[29] with Long Short Term Memory (LSTM) cells [11]. The RNNs can be trained to maximize the following objective function:

$$L(Y) = \log p(Y|X). \tag{3}$$

Unbalanced distribution between blank and non-blanks. Based on the mapping function F, blanks are inserted to label sequence to make paths. Blank can exist in almost every training data as long as the sequence length T is greater than label length T^y. Besides, the positions of blanks in paths are more flexible compared to non-blank symbols, which makes the unbalanced problem worse. The unbalanced distribution between blank and non-blank symbols leads to the non-blank symbols are not aligned to the input feature sequence sufficiently. This also means the model is underfitting the non-blanks.

2.3 Variational Connectionist Temporal Classification

As the unbalanced distribution between blank and non-blank symbols in CTC is inevitable, computing the distributions of blank and non-blank symbols separately may reduce the influence. So we propose to change the unified distribution to hierarchy distribution. Specifically, we put forward the following objective:

$$L(Y) = \log p(Y|X) = \log \sum_{O_b} p(O_b|X) p(Y|O_b, X). \tag{4}$$

The Eq 4 shows the model first determines the blank, then outputs the non-blanks. The blank outputs determine the alignment paths between label and input sequence, which is vital for the unsupervised alignments. Given the posterior distribution of blank ($p(O_b|X, Y)$), the model should learn the alignment for non-blanks better. Thus, we propose to approximate the posterior of blank instead of learning the likelihood ($p(O_b|X)$). With the help of variational inference optimization strategy, we try to optimize the evidence lower bound (ELBO)[16] of $L(Y)$. In this paper, we define the ELBO as the Var-CTC loss. It is defined as:

$$L_{var-ctc}(Y) = E_{q_\psi(O_b|X,Y)}[\log p(Y|O_b, X)] - KL(q_\psi(O_b|X, Y)||p_\beta(O_b|X)). \tag{5}$$

It is composed of three different terms, likelihood $p(Y|O_b, X)$, prior $p_\beta(O_b|X)$ and posterior $q_\psi(O_b|X, Y)$. These three terms can be parameterized by three different layers, like multilayer perceptrons (MLPs) and RNN. Fig. 2 shows the schematic diagram of the Var-CTC based model. For parameter efficiency, we utilize the MLP to represent the posterior and prior distributions in this paper. These three terms are described below:

Likelihood. Given the approximate posterior for blanks, the likelihood aggregates the probabilities of all possible paths. Different with the vanilla CTC, at each timestep, the categorical distribution form x^t to A^* is calculated as:

$$p(o^t|o_b^t, x^t) = \begin{cases} q_\psi(o_b^t|x^t, Y), & \text{if } o^t = \text{``–''}, \\ p(o_c^t|x^t) \cdot (1 - q_\psi(o_b^t|x^t, Y)), & \text{otherwise}, \end{cases} \tag{6}$$

where $p(o_c^t|x^t) = softmax(x^t W_c)$ is the class distribution over all non-blank symbols in the vocabulary. $W_c \in \mathbb{R}^{d \times |A|}$ is the matrix parameter.

Once we get the output possibilities at each timestep, the function F maps the paths to sequence output like CTC. During inference time when the label sequence is not available, the prior term is used to replace posterior approximator to output the blank distribution.

Prior. The prior $p_\beta(O_b|X)$ is formulated as a sequence of Bernoulli distributions. Given feature sequence, it computes the distribution of blank at all the timesteps. For efficiency, we directly convert the feature at every timestep to distributions with one feed-forward layer. It is computed as follows:

$$p_\beta(o_b^t|x^t; \beta) = \sigma(x^t w_p), \tag{7}$$

where $\sigma(\cdot)$ is the sigmoid function with $w_p \in \mathbb{R}^d$ being vector parameter.

Approximate posterior. The posterior distribution $q_\psi(O_b|X, Y)$ follows the similar architecture as the prior. The main difference lies in the fact that posterior approximator is aware of the label sequence Y, therefore outputting more accurate distributions. In order to incorporate the label sequence, we embed each symbol in vocabulary to a random initialized vector. These embeddings are learned with other parameters in the network together. For a particular label sequence, we get the label representation $\bar{Y}$ by mean pooling operation on time axis. The posterior is then computed as follows:

$$q_\psi(o_b^t|x^t, Y; \psi) = \sigma((f(x^t) \circ \bar{Y}) w_a), \tag{8}$$

where function f is one feed-forward layer and it converts dimension of x^t to the same dimension of $\bar{Y}$. The $\circ$ denotes the Hadamard product. $w_a \in \mathbb{R}^d$ is the vector parameter.

Optimization. The loss function has two terms, which can be optimized jointly. The first term in Eq 5 is motivated to get the target label based on blank distribution. Optimizing this term can also help the approximate posterior to obtain accurate blank output distribution. The second term is the KL-divergence between the prior distribution and the approximated posterior distribution, which is motivated to push the prior distribution towards the posterior distribution. The values of $q_\psi(O_b|X, Y)$ can be directly utilized to compute

$p(Y|O_b, X)$ based on the forward-backward algorithm[4] and the mapping function F in vanilla CTC. Thus, the distributions $q_\psi(O_b|X, Y)$ and $p(Y|O_b, X)$ in the first term can be optimized based on the forward-backward algorithm like the vanilla CTC loss. The second term is the KL divergence between two distributions. The gradients of the second term can also be calculated analytically. So the loss can be optimized with any gradient based optimization algorithm.

The O_b can also be modeled as discrete values similar to the Sampled CTC[32]. It means their values are sampled binary values based on q_ψ. In this way, as O_b is not differentiable with respect to q_ψ, the REINFORCE algorithm[33] can be used to tackle this problem. One advantage of this modeling way is it can speed up the training process since the forward-backward algorithm is not needed. The disadvantage is that variance reduction technique for the REINFORCE based optimization should be considered. We leave this line for future work.

Complexity Analysis. The time complexity of CTC and Var-CTC forward-backward dynamic programming is $O(T)$. Compared to CTC, the mainly added computation is the posterior distribution, which is also $O(T)$. Since the forward and backward variable are kept for gradient computing, the space complexity of CTC and Var-CTC is $O(TT^y)$.

3 Experimental Results

In our experiments, two tasks are employed to evaluate the effectiveness of Var-CTC, including handwritten text recognition and scene text recognition. Besides, we also try to directly maximize the marginal likelihood $p(Y|X) = \sum_{O_b} p(O_b|X)p(Y|O_b, X)$ using only the prior and likelihood model following [8], which enables us to understand the superiority of introducing an approximate posterior. We call this objective as Mml-CTC.

3.1 Scene Text Recognition

Convolutional Recurrent Neural Network (CRNN) [30] is utilized as the feature extraction network. We compare our method with CRNN-CTC [30] and CRNN-EnEs-CTC[20] models. All models have the same feature extraction network.

Evaluation Metrics. There are two evaluation metrics in this experiment. The first one is the sequence accuracy, which is in accordance with the experiments setup of [30, 20]. Sequence accuracy means the percentage of test images that are recognized totally correct.

The second one is the Precision-Recall curve. The confidence score is computed based on the greedy decode method. Greedy decode is the widely used decode method in scene text recognition [30, 20]. It decodes the outputs by choosing the most feasible path, which is the concatenation of the most probable labelling for every timestep. The confidence score corresponding to this path is calculated as:

$$p(\pi^*|x) = \prod_{i=1}^{T} \max p(o^t|x^t). \tag{9}$$

Previous studies evaluate the model on benchmark datasets on the resized images. For example, both [30, 20] resize the test images to 100 x 32. Thus, we plot the Precision-Recall curves based on the resized images.

Datasets. We train our models with the large scale synthetic dataset Synth90K [14] and test on four real-world benchmark datasets following [30, 20]. Synth90K contains 8 million training images and 1 million test images. All the images are generated by a synthetic data engine using a 90k word dictionary. These four real-world test datasets are ICDAR 2003 (IC03) [21], ICDAR 2013 (IC13) [17], IIIT5kword (IIIT5K) [25] and Street View Text (SVT) [15]. IC03 test set consists of 251 full scene images and 860 cropped image patches containing words. IC13 extends IC03 and contains 1015 cropped word images from real scenes. In the experiments, only words with alphanumeric characters and at least three characters are considered. There are 860 and 857 test images are utilized for IC03 and IC13 respectively. IIIT5k contains 3,000 cropped word images downloaded from Google Image Search. SVT contains 647 word images cropped from 249 street-view images. The images are collected from Google Street View.

In the experiment, we use the 8 million training images of Synth90K as the training data and the 1 million test images as the validation data. The maximum iteration step is 1,600,000, which is roughly 50 epochs. We validate the models every 10K iterations on the validation set. The best models are picked based on the sequence accuracy performance on the validation set.

Implementation Details. We use Tensorflow [1] to implement all the models. Our Tensorflow based implementation has two differences compared to the implementation of [30]. The first one is the different padding way in the third and fourth maximum pooling layers. We use the 0 x 0 padding in Tensorflow compared to 0 x 1 padding in Pytorch[1]. The second difference is that we add dropout [10] with probability 0.1 after convolutional layers except the first and the last ones. Because we find dropout improves performances. In order to accelerate the training process, all the images are resized to 100 x 32.

We take the outputs of the last BLSTM layer as feature sequence for the input image. In Var-CTC, the dimension of label embedding is set to 50. In order to prevent overfitting, we also add dropout with probability 0.5 for the pooled label embedding. We use Adam [18] to train all the models with batch size set to 256. The learning rate is fixed at 0.001 for the CTC and Mml-CTC based models. For Var-CTC based model, we set the learning rate to 0.0005.

Comparison results. The sequence accuracy comparisons are shown in Table 1. Compared to the CRNN-CTC model[30], our implementation shows clear improvements on all the four datasets, which are 0.3%, 2.9%, 1.2% and 0.8% respectively. The improvements mainly come from the added dropout operation in our implementation. Compared to the CRNN-EnEs-CTC model, our implementation does not show advantage on sequence accuracy performance.

The comparison between CRNN-Mml-CTC and CRNN-CTC(ours) shows that the CRNN-Mml-CTC model improves 1.9% and 0.2% on IIIT5K and SVT respectively. And it also shows that CRNN-Mml-CTC brings down the sequence

[1] Tensorflow does not support 0 x 1 padding for MaxPooling operation.

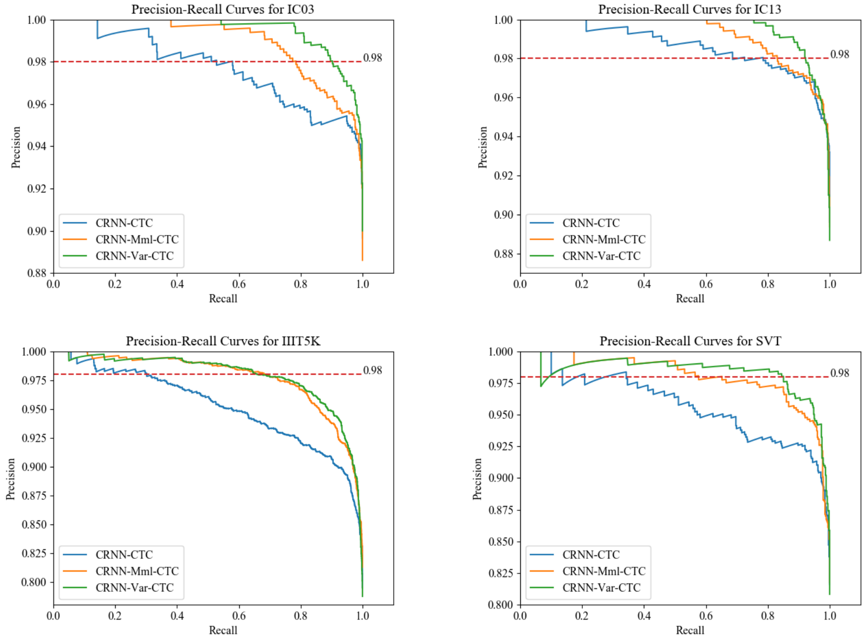

Fig. 3: Precision-Recall curves on four real-world datasets. The red dotted line is used to indicate the precision value equals to 98%.

accuracy with 0.7% and 0.2% on IC03 and IC13 respectively. While the CRNN-Var-CTC achieves the best performance on IC03 dataset compared to the CTC and Mml-CTC models, it has no advantage for the other three datasets. We can conclude the proposed Mml-CTC objective is comparable with vanilla CTC loss and the Var-CTC loss has slightly inferior performance when evaluating sequence accuracy.

The Precision-Recall curves of the compared models on the four datasets are shown in Fig. 3. The comparison results also show Mml-CTC and Var-CTC have clear improvements compared to the vanilla CTC and Var-CTC gets the best performances. These means the proposed Var-CTC and Mml-CTC perform

Table 1: Comparisons of SeqAcc on the four real-world datasets.

Method	IC03	IC13	IIIT5K	SVT
CRNN-CTC[30]	89.4	86.7	78.2	80.8
CRNN-EnEs-CTC[20]	**92.0**	**90.6**	**82.0**	80.6
CRNN-CTC(ours)	89.7	89.6	79.4	81.6
CRNN-Mml-CTC	88.6	89.4	81.3	**81.8**
CRNN-Var-CTC	90.0	88.7	78.7	80.8

Table 2: Comparisons of recall performances when all the models maintain 98% precision on the four real-world datasets.

Method	IC03	IC13	IIIT5K	SVT
CRNN-CTC	58.0	78.5	30.2	34.3
CRNN-Mml-CTC	78.5	83.2	**68.7**	57.7
CRNN-Var-CTC	**90.3**	**92.1**	66.4	**85.1**

better in average precision (AP). We also compare the recall performances when the models maintain 98% precision. The comparison results are shown in Table 2. Under the same condition, the Var-CTC can improve the recall with 31.7%, 13.6%, 36.2% and 50.8% respectively. In practical use, the proposed Var-CTC can recall more samples compared the vanilla CTC.

Table 3: Comparisons with Attention based model.

Method	SeqAcc	AP	Recall@Precision=98%
ASTER[31]	**86.21**	98.37	69.84
CTC	82.37	98.01	67.57
Mml-CTC	83.83	98.59	77.21
Var-CTC	81.65	**98.64**	**81.92**

Comparison with Attention based Method. We compare our proposed methods with ASTER [31]. ASTER is an attention based model and its well-trained models are public available. We take the same experiment setup of ASTER. The only difference is that all our CTC based models only utilize the ResNet based backbone, without the Spatial Transformer[15] based thin-plate spline (TPS) transformation network. There are seven testing datasets in ASTER. Due to limited space, we replace Precision-Recall curve with AP metric and the comparison in Table 3 is based on collection of all the seven datasets. The comparison results also show Var-CTC has the best AP or Precision-Recall performance, especially the recall performance at a high precision level.

3.2 Offline Handwritten Text Recognition

To verify the generalization capability of our method, we further evaluate our method on offline handwritten text recognition. Compared to scene text recognition, offline handwritten text recognition problem is highly complicated and challenging to solve. In the experiment, we follow the experiment setup of [34].

Datasets. The public handwritten datasets IAM [23] is used in this experiment. IAM is a handwritten text dataset, with 647 writers. It is partitioned into writer-independent training, validation and test partitions, where each partition contains 46945, 7554 and 20306 correctly segmented words respectively.

Evaluation Metrics. Three metrics are used to evaluate the handwritten text recognition model. The first two are the Character Error Rate (CER) and the Word Error Rate (WER). CER is defined as the Levenstein distance between the predicted and real character sequence of the word. WER denotes the percentage of words improperly recognized. For CER and WER, small values indicate better performance. The last metric is the Precision-Recall curve. We also use greedy decoding to decode the outputs. The confidence score of each word is computed based on Eq 9.

Implementation Details. Different with the experiment in scene text recognition, we use a 25-layer residual network [9] as convolutional feature extractor.

As IAM is much smaller than Synth90K, we stop the training at 20k iterations. The other setups are the same with scene text recognition task. We name the feature extraction network as ResNet to distinguish the CRNN in scene text recognition experiment.

Comparison results. The comparison results are shown in Table 4 and Figure 4. We also compare our method with the state-of-the-art approaches [34]. For WER and CER metrics, the proposed Var-CTC has no advantage compared to the vanilla CTC. However, for Precision-Recall curve metric, the proposed Var-CTC and Mml-CTC show strong performances compared to vanilla CTC.

Table 4: Comparisons of WER and CER on IAM.

Method	WER	CER
zhang2019sequence[34]	**22.2**	**8.5**
ResNet-CTC	23.8	9.53
ResNet-Mml-CTC	23.6	9.35
ResNet-Var-CTC	24.0	9.52

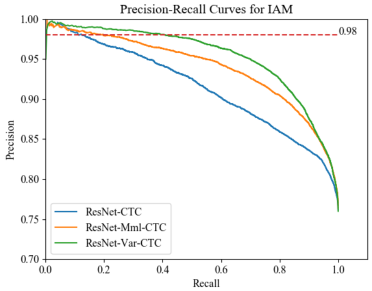

Fig. 4: Precision-Recall curve performance comparisons on IAM.

3.3 Further Analysis

We first analyze the learning signals of non-blanks to check whether their learning is enhanced. Then we analyze whether the enhanced learning lead to a better alignment, especially for the non-blanks. We try to explain the reason for the improved Precision-Recall curve performance. The badcase analysis and the label embedding visualization for Var-CTC are also included in this part. At last, we give the exact comparisons of the space and time for the proposed models. All these analysises are based on scene text recognition task.

Gradient Signal Analysis. Fig. 5 shows one example about the evolution of gradient signals for non-blank symbols. The experiment is based on the Synth5K[20] dataset. Synth5K is a small dataset with 5K training data sampled randomly from Synth90K. Both the Var-CTC and CTC based models predict correctly for the chosen case. We can observe four points from the comparisons. Firstly, at the initial training stage, the gradient signals for Var-CTC are less than the CTC model; Secondly, the gradient signals of the CTC model decay much more quickly than the Var-CTC model; Thirdly, the gradients of CTC model quickly focus to several isolated points; At last, in the middle (30 epochs) and late (50 epochs) training periods, the gradients of Var-CTC are much greater

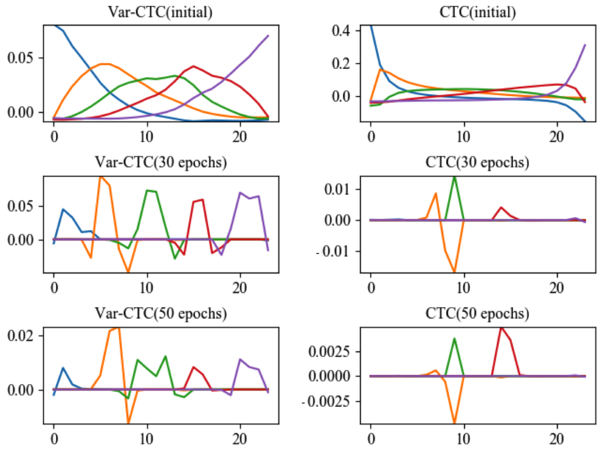

Fig. 5: The evolution of gradient signals for non-blank symbols. The input image is the first example in Fig. 6. The horizontal axis denotes the position in the image and colors represent different labels.

than the CTC model. From the comparison, we can see the learning of non-blank symbols in Var-CTC model is more stable and sufficient.

Alignment Analysis. We depict the output distributions of six examples for CTC and Var-CTC based models in Fig. 6. The figure shows both the outputs of the two models are spiky, where blank dominates the outputs. We also depict the second hierarchy outputs for Var-CTC model. All the six examples show the distribution of $p(o_c^t|x^t)$ aligns to the corresponding non-blanks sufficiently, even for the cases with irregular shapes. The alignment visualization can be explained by the gradient signal visualization in Fig. 5, which is better learning signal leads to better alignment.

Confidence Score Analysis. We can also find the reason why the confidence score of the CTC model is not discriminative enough. The "SHARE" example in Fig. 6 shows there are two successive timesteps that the CTC model is aligned to the label "A". At the first timestep, it outputs "A" with probability close to 1.0. While at the successive timestep the model outputs "A" with probability close to 0.3 and blank with probability close to 0.7. It shows there is ambiguity at the second timestep between "A" and blank. Although the "SHARE" example is a clear and easy to recognize image visually, the confidence score of it keeps close to some hard examples (for example, the third example in Fig. 6). For the CTC model, the prediction confidence score of "SHARE" is less than "KEEPERS".

The "TRUSTPASS" example in Fig. 6 shows both the CTC and Var-CTC are influenced by the image content after the second "S". However, for this wrong decision, the CTC model still outputs a high level confidence score at the last timestep, which is close to probability 1.0. While the Var-CTC model is not sure which character it looks like and output character with low confidence score. As

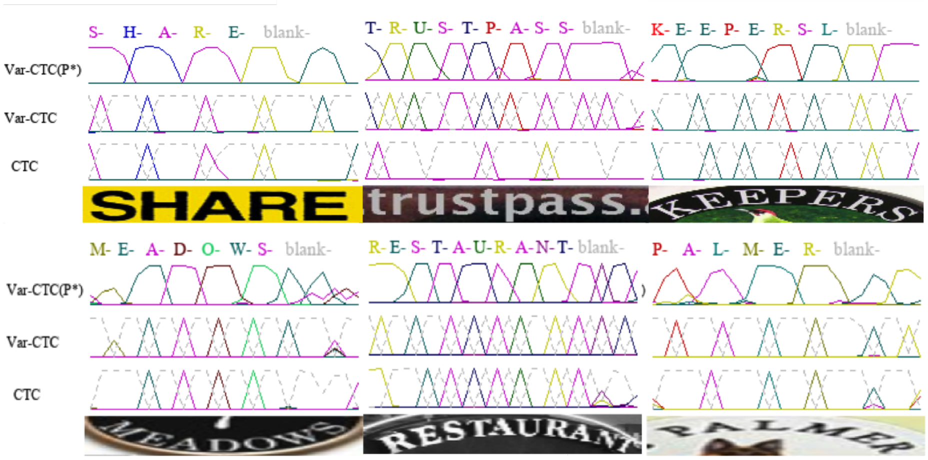

Fig. 6: Outputs visualizations of six examples. P^* represents the second hierarchy outputs $(p(o_c^t|x^t)$ in Eq 6.

these cases show, the confidence score of the CTC model is not discriminative enough. It is difficult to set the threshold in practical use. And the proposed Var-CTC can relieve this problem with the improved confidence score.

Table 5: Error analysis for the IIIT5K datasets.

Method	Replace	Delete	Insert
CRNN-CTC	49%	14%	37%
CRNN-Var-CTC	40%	19%	41%

Error Analysis. We roughly classify the prediction errors into three types based on the sequence lengths of the labels and predictions. These three types are "Replace", "Delete" and "Insert". "Replace" means element in the label sequence is replaced by other symbols in the prediction sequence. "Insert" means the model predicts extra symbols compared to the label. We can see the blank outputs contribute more to the "Delete" and "Insert" types as those two types show segmentation error exists. The alignments between non-blank symbols and image contribute more about the "Replace" type. Table 5 shows the error analysis on IIIT5K datasets. Compared with the CTC model, the ratio of the "Replace" error is declined for the Var-CTC model. This proves the alignments of the non-blank symbols are improved. The statistics also shows the approximate posterior of blank should be improved.

Label Embedding Visualization. We visualize the label embeddings learned by Var-CTC in Fig 7. For better visualization, we do not show the embeddings of 26 letters in English alphabet. The learned embeddings nicely reflect the clustering structures among the characters (the "6", "8" and "9" have similar shapes).

It means the added label embedding has positive effect to the model learning, which can explain the superiority of Var-CTC compared to Mml-CTC.

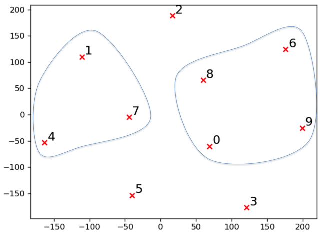

Fig. 7: Visualization of the label embeddings learned by Var-CTC using t-SNE[22]. Only the 10 number characters are shown.

Table 6: Comparisons of Parameters and FLOPS.

Model	#Parameters	#FLOPS
CRNN-CTC	8,720,165	17,436,510
CRNN-Mml-CTC	8,720,165	17,436,511
CRNN-Var-CTC	8,747,765	17,491,816

Space and Time Comparisons. Table 6 lists the exact number of parameters and FLOPS for models in 3.1. All the MLPs in Fig.2 are one layer in our implementation and they are used to transform features to the desired dimensions. Mml-CTC has exactly the same number of parameters with CTC. Compared to Mml-CTC, Var-CTC adds the approximated posterior. The table shows Var-CTC adds extra 0.3% more parameters and 0.3% more FLOPS compared to CTC, which is neglectable.

4 Conclusion

We proposed the variational CTC to improve CTC for enhancing the learning of the non-blank symbols. The proposed Var-CTC first determines whether to output a specific label or not and then learns the alignment of the non-blank symbols with the input feature sequence. With the hierarchy output, the influence of imbalanced distribution between blank and non-blanks is relieved. With the enhanced learning of non-blanks, our model can output more reliable confidence scores, which is important in practical use. Experiments on scene text recognition and offline handwritten text recognition tasks show the proposed Var-CTC improves the Precision-Recall curve performances significantly. Under the same condition, the Var-CTC can improve the recall performance with a large margin on four five-world benchmark datasets. Qualitative analysis also shows the effectiveness of the proposed method. As for future work, we plan to try variational inference techniques such as [19, 27] to improve the approximate posterior distributions.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp. 265–283 (2016)
2. Cui, R., Liu, H., Zhang, C.: Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7361–7369 (2017)
3. Feng, X., Yao, H., Zhang, S.: Focal ctc loss for chinese optical character recognition on unbalanced datasets. Complexity **2019** (2019)
4. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
5. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: International conference on machine learning. pp. 1764–1772 (2014)
6. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence **31**(5), 855–868 (2008)
7. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6645–6649. IEEE (2013)
8. Guu, K., Pasupat, P., Liu, E.Z., Liang, P.: From language to programs: Bridging reinforcement learning and maximum marginal likelihood. arXiv preprint arXiv:1704.07926 (2017)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
12. Hu, W., Cai, X., Hou, J., Yi, S., Lin, Z.: Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In: AAAI. pp. 11005–11012 (2020)
13. Huang, D.A., Fei-Fei, L., Niebles, J.C.: Connectionist temporal modeling for weakly supervised action labeling. In: European Conference on Computer Vision. pp. 137–153. Springer (2016)
14. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: Advances in neural information processing systems. pp. 2017–2025 (2015)
16. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine learning **37**(2), 183–233 (1999)
17. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1484–1493. IEEE (2013)

18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
20. Liu, H., Jin, S., Zhang, C.: Connectionist temporal classification with maximum entropy regularization. In: Advances in Neural Information Processing Systems. pp. 831–841 (2018)
21. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R., Ashida, K., Nagai, H., Okamoto, M., Yamamoto, H., et al.: Icdar 2003 robust reading competitions: entries, results, and future directions. International Journal of Document Analysis and Recognition (IJDAR) **7**(2-3), 105–122 (2005)
22. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
23. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition **5**(1), 39–46 (2002)
24. Miao, Y., Gowayyed, M., Metze, F.: Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). pp. 167–174. IEEE (2015)
25. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors (2012)
26. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
27. Rezende, D.J., Mohamed, S.: Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770 (2015)
28. Sak, H., Senior, A., Rao, K., Irsoy, O., Graves, A., Beaufays, F., Schalkwyk, J.: Learning acoustic frame labeling for speech recognition with recurrent neural networks. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 4280–4284. IEEE (2015)
29. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing **45**(11), 2673–2681 (1997)
30. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence **39**(11), 2298–2304 (2016)
31. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**, 2035–2048 (2018)
32. Variani, E., Bagby, T., Lahouel, K., McDermott, E., Bacchiani, M.: Sampled connectionist temporal classification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4959–4963. IEEE (2018)
33. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4), 229–256 (1992)
34. Zhang, Y., Nie, S., Liu, W., Xu, X., Zhang, D., Shen, H.T.: Sequence-to-sequence domain adaptation network for robust text image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2740–2749 (2019)