MACRO-BLOCK DROPOUT FOR IMPROVED TRAINING OF END-TO-END SPEECH RECOGNITION MODELS

Chawoo Kim[1]*, Sathish Indurti*[1]*, Jinhwan Park*[1]*, and Wonyong Sung*[2]

Samsung Research[1], Seoul, South Korea
Seoul National University[2], Seoul, South Korea
{chanw.com, s.indurti, jh0354.park}@samsung.com, wysung@snu.ac.kr

ABSTRACT

This paper proposes a new regularization algorithm referred to as a *macro-block dropout*. The overfitting issue has been a difficult problem in training large neural network models. The dropout technique has proven to be simple yet very effective for regularization by preventing complex co-adaptations on training data. In this work, we observe that among hidden outputs, the correlations between geometrically close elements are usually stronger than those between distant elements. Motivated by this observation, we define a *macro-block* that contains multiple elements of the hidden output layer in order to reduce co-adaptations more effectively. Rather than applying dropout to each element, we apply random dropout to each *macro-block*. In our experiments using Recurrent Neural Network-Transducer (RNN-T) and Attention-based Encoder Decoder (AED) models, this simple algorithm has shown relatively 4.33 % and 5.99 % Word Error Rate (WER) improvements over the conventional dropout approach on LibriSpeech `test-clean` and `test-other`. The *Keras* layer implementation of this algorithm will be released as open-source.

Index Terms: neural-network, regularization, macro-block, dropout, end-to-end speech recognition

1. INTRODUCTION

Deep learning technologies have significantly improved speech recognition accuracy recently [1]. There have been series of remarkable changes in speech recognition algorithms during the past decade. These improvements have been obtained by the shift from Gaussian Mixture Model (GMM) to the Feed-Forward Deep Neural Networks (FF-DNNs), FF-DNNs to Recurrent Neural Network (RNN) such as the Long Short-Term Memory (LSTM) networks [2]. Thanks to these advances, voice assistant devices such as Google Home [3], Amazon Alexa and Samsung Bixby are widely used at home environments.

Recently tremendous amount of research has been conducted for switching from a conventional speech recognition system consisting of an Acoustic Model (AM), a Language Model (LM), and a decoder based on a Weighted Finite State Transducer (WFST) to a complete end-to-end all-neural speech recognition system [4]. A large number of these end-to-end speech recognition systems are based on the Attention-based Encoder Decoder (AED) [4] and the Recurrent Neural Network-Transducer (RNN-T) [5] algorithms. These complete end-to-end systems have started outperforming conventional WFST-based decoders for large vocabulary speech recognition tasks [6]. Further improvements in these end-to-end speech recognition systems have been possible thanks to a better choice of target units such as Byte Pair Encoded (BPE) and *unigram language model* [7] subword units, and an improved training methodologies such as Minimum Word Error Rate (MWER) training [8].

In training such all neural network structures, overfitting has been a major issue. For improved regularization in training, various approaches have been proposed [9]. Data-augmentation has been also proved to be useful in improving model training [3, 10, 11]. The dropout approach [12] has been applied to overcome this issue in which both the input and the hidden units are randomly dropped out to regularize the network. In the case of the input dropout, the input feature elements are masked with a certain fixed probability of p, and the remaining input feature elements are scaled up by $1.0/(1.0-p)$. This dropout approach has inspired a number of related approaches [13, 14, 15]. In *DropBlock* [16], it has been argued that dropping out at random is not effective in removing semantic information in input images because nearby activations contain closely related information. Motivated by this, they apply a square mask centered around each zero position.

In this paper, we present a new regularization algorithm referred to as *macro-block dropout* We conjecture that in the hidden outputs of neural network layers, the correlations between geometrically close elements are stronger than those between distant elements. We define a *macro-block* that contains multiple elements of the hidden output layer in order to reduce co-adaptations more effectively. Rather than applying dropout to each element, we apply random dropout to each *macro-block*.

Our *macro-block dropout* approach is motivated by an idea similar to *DropBlock* [16]. However, our *macro-block dropout* is unique in the following aspects. First, *DropBlock* is only targeted for Convolutional Neural Networks (CNNs). They have not considered applying their algorithm to the output of RNNs. In our work, we focus on improving end-to-end speech recognition models based-on RNNs. As will be discussed Sec. 5, we observe that the effect of *Macro-block dropout* is slightly better if the *macro-block dropout* pattern is kept the same across the time. This masking pattern is illustrated in Fig. 2b. To the best of our knowledge, our work is the first in applying bigger chunks of masks consisting of multiple elements to RNNs. Second, instead of applying the scaling approach used in the original dropout in (1) or the scaling method based on the count of masks of ”one” values in *DropBlock*, we scale the output using the equation (5). As will be discussed in Sec. 3.2, this scaling approach is more effective than other scaling approaches for the *macro-block dropout* case. In our experiments using an RNN-T [5] in Sec. 5, this simple algorithm has shown a quite significant improvement over the conventional dropout approach.

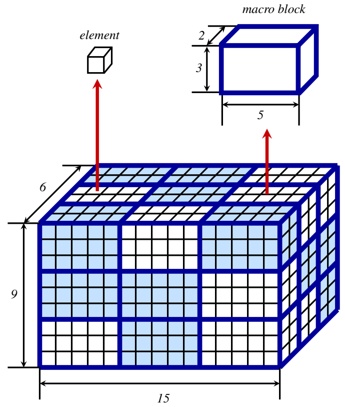


Fig. 1: The relationship between a 3-dimensional tensor represented by a space, its elements, and *macro-blocks*: This tensor has the shape of $\mathbb{d}_{\mathbf{x}} = (15, 9, 6)$. After re-partitioning by *macro-blocks*, it has the shape of $\mathbb{d}_{(\text{par})} = (3, 3, 3)$. The shape of each *macro-block* is $(5, 3, 2)$. *Macro-blocks* to be dropped out are marked in blue color.

2. RELATED WORKS

Dropout is a simple regularization technique to alleviate the overfitting problem by preventing co-adaptations [12]. When the shape of the output of a neural network layer is $\mathbb{d}_{\mathbf{x}}$, we create a random mask tensor $\mathbf{m}$ with the same shape $\mathbb{d}_{\mathbf{x}}$. Each element $m \in \mathbf{m}$ follows the Bernoulli distribution $m \sim \mathit{Bernoulli}(1 - p)$, where p is the dropout rate. Given an input $\mathbf{x}$, the dropout output $\mathbf{x}_{\text{out}}$ is obtained by the following equation:

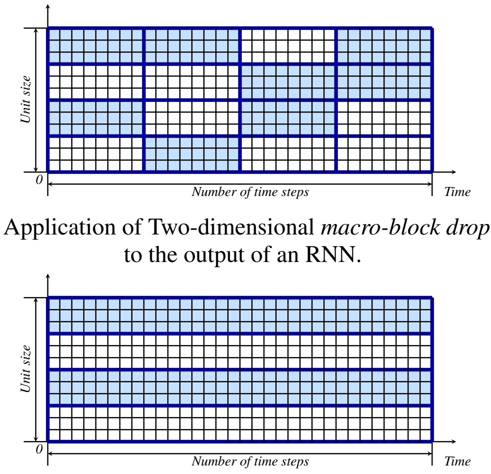
$$\mathbf{x}_{\text{out}} = \frac{\mathbf{x} \odot \mathbf{m}}{1 - p}, \tag{1}$$

where $\odot$ is a Hadamard product. The scaling by $\frac{1}{1-p}$ is applied to keep the sum of elements the same through this masking process. Dropout has been turned out to be especially useful in improving the training of dense network models for image classification [17], speech recognition [18], and so on. This dropout approach inspired many other related approaches such as *DropConnect* [13], *droppath* [14], shake-shake [15], *ShakeDrop* [19], and *DropBlock* [16] regularizations.

The *DropBlock* approach described in [16] has similar motivation to our *macro-block dropout* approach. In [16], they observe that dropping out a tensor containing an input image at random is not effective in removing semantic information because nearby activations contain closely related information. They conjecture that the same reasoning is valid for hidden layers. In *DropBlock*, the zero position is selected in the same way as the *DropOut*. However, for each zero position, a spatial square mask with the center at that zero position is created. It has been reported that *DropBlock* significantly outperforms the baseline *dropout*.

3. MACRO-BLOCK DROPOUT

In this section, we rigorously define *macro-blocks* in different dimensions in Sec. 3.1. We describe the *macro-block dropout* algorithm in detail in Sec. 3.2.



(a) Application of Two-dimensional *macro-block dropout* to the output of an RNN.

(b) Application of One-dimensional *macro-block dropout* to the output of an RNN.

Fig. 2: The *macro-block dropout* approach applied to the output of a Recurrent Neural Network (RNN) layer: (2a) Two-dimensional and (2b) One-dimensional *macro-block dropout* cases. Each tiny rectangle defined by the grid corresponds to each element of the RNN output. Larger rectangular chunks are *macro-blocks*. Region in the light blue color represent *macro-blocks* to be dropped out.

3.1. Definition of a macro-block

Let us consider a D-dimensional tensor $\mathbf{x}$ that is output activations of a neural-network layer. Suppose that the shape of $\mathbf{x}$ is given by:

$$\mathbb{d}_{\mathbf{x}} = (N_1, N_2, \cdots, N_D). \tag{2}$$

Macro-blocks are constructed by equally partitioning this space defined by (2) along each dimension into the following shape:

$$\mathbb{d}_{(\text{par})} = (P_1, P_2, \cdots, P_D), \tag{3}$$

with the following constraint:

$$P_d \leq N_d. \qquad \text{for } 1 \leq d \leq D. \tag{4}$$

Fig. 1 shows an example of a three-dimensional case. In this example, the shape of $\mathbf{x}$ is $\mathbb{d}_{\mathbf{x}} = (15, 9, 6)$. The region defined by each grid corresponds to each element of a tensor $\mathbf{x}$. After re-partitioning this space defined by $\mathbb{d}_{\mathbf{x}}$ into *macro-blocks*, we observe that $\mathbb{d}_{(\text{par})} = (3, 3, 3)$ as shown in Fig. 1. The shape of each macro block is $(6, 3, 2)$ in this example.

This *macro-block* concept can also be applied to Recurrent Neural Networks (RNNs). Fig. 2 illustrates *macro-blocks* applied to the output of an RNN. The output of an RNN layer has the shape of $(\text{unit\_size}, \text{number\_of\_time\_steps})$. Fig. 2a shows the case when this two-dimensional region is partitioned by $\mathbb{d}_{(\text{par})} = (4, 4)$. We may apply this two-dimensional *macro-blocks* to RNNs. However, as observed with the baseline dropout [20], we observe that speech recognition accuracy is better if mask patterns remain the same along the time axis. If the *macro-block* masks do not change along the time axis, the mask pattern becomes a one-dimensional case as shown in Fig. 2b. The WERs using an RNN-T model with the one-dimensional and the two-dimensional *macro-block dropout* approaches are summarized in Table 1. The RNN-T model and experimental configuration for obtaining these WERs are described

Algorithm 1 Macro-block Dropout

```
Input: output activations of a layer: $\mathbf{x}$, the shape after partitioning: $\mathbb{d}_{(\text{par})}$, dropout_rate $p$, mode
if mode == Inference then
    return $\mathbf{x}$
end if
Creates a random tensor $\mathbf{r}$ with a shape of $\mathbb{d}_{(\text{par})}$:
    For each element $r$ of $\mathbf{r}$, $r \sim Bernoulli(1 - p)$.
Creates a masking tensor $\mathbf{m}$ by resizing $\mathbf{r}$ using the nearest-neighbour method to match the dimension of $\mathbf{x}$.
Applies the mask:
    $\mathbf{x}_m = \mathbf{x} \odot \mathbf{m}$.
Obtains the output $\mathbf{x}_{\text{out}}$ by scaling $\mathbf{x}_m$ :
    $\mathbf{x}_{\text{out}} = \left| \frac{\sum_{\text{all elements}} \mathbf{x}}{\sum_{\text{all elements}} \mathbf{x} \odot \mathbf{m}} \right| \mathbf{x}_m$.
```

in detail in Sec. 5. From this result, we conclude that the one-dimensional *macro-block dropout* approach is more effective than two-dimensional approach for RNNs. In obtaining this result, we choose the partition shape of $\mathbb{d}_{(\text{par})} = (4)$ for the one-dimensional case and $\mathbb{d}_{(\text{par})} = (4, 4)$ for the two-dimensional case. These partition shapes result in the best WERs for one- and two- dimensional *macro-block dropout* cases respectively in our experiments on the *LibriSpeech* corpus experiments.

Table 1: Word Error Rates (WERs) with the RNN-T model shown in Fig. 3a using the one-dimensional *macro-block dropout* of $\mathbb{d}_{(\text{par})} = (4)$, and the two-dimensional *macro-block dropout* of $\mathbb{d}_{(\text{par})} = (4, 4)$. In these experiments, the dropout rate of 0.2 is used since the best WER in each case is obtained at this rate.

Test Set	Baseline Dropout	Macro-Block Dropout	
		1-D $\mathbb{d}_{(\text{par})} = (4)$	2-D $\mathbb{d}_{(\text{par})} = (4, 4)$
`test-clean`	**3.95** %	**3.78** %	3.92 %
`test-other`	**12.23** %	**11.48** %	11.50 %
Average	**8.09** %	**7.63** %	7.71 %

3.2. Application of dropout to macro-blocks

Having defined the required terms in Sec. 3.1, we proceed to explain the algorithm in detail in this section. The entire algorithm is summarized in Algorithm 1. During the inference time, *macro-block dropout* is not applied as the original dropout. During the training time, we create a random tensor $\mathbf{r}$ whose shape is $\mathbb{d}_{(\text{par})}$. This tensor is created from the *Bernoulli* distribution with the probability of one given by $1 - p$, where p is the dropout probability. This $\mathbf{r}$ is then resized to match the shape of the input $\mathbf{x}$. For simplicity, this resize operation is performed using the nearest-neighborbood approach.

The scaling factor r is given by the following equation:

$$r = \left| \frac{\sum_{\text{all elements}} \mathbf{x}}{\sum_{\text{all elements}} \mathbf{x} \odot \mathbf{m}} \right|. \tag{5}$$

We apply the absolute value operation in (5), because the sign of the numerator and the denominator of (5) may be different when $\mathbf{x}$ is the output of an RNN such as an LSTM or a GRU. More specifically, the hidden output of an LSTM is given by the following equation [2, 21]:

$$\mathbf{h}_{[m]} = \mathbf{o}_{[m]} \odot \sigma_h(\mathbf{c}_{[m]}), \tag{6}$$

where m is a time index, $\odot$ is the Hadamard product, $\sigma_h(\cdot)$ is the hyperbolic tangent function, $\mathbf{o}_{[m]}$ is the output-gate value, and $\mathbf{c}_{[m]}$ is the cell value, respectively. From (6), it is obvious that $\mathbf{h}_{[m]}$ may have both positive and negative values, since the range of of σ_h is between -1 and 1. In our speech recognition experiments, it is observed that performance is slightly worse if this absolute value operation is not applied in (5).

We observe that the scaling in (5) is more effective than a simple scaling of $\frac{1}{1-p}$ used in the baseline dropout in (1). Table 2 summarizes WERs obtained with the conventional scaling of $\frac{1}{1-p}$ and the scaling in (5) on the *LibriSpeech* `test-clean` and `test-other` sets. We use an RNN-T model that will be described in Sec. 5. For *macro-block dropout*, we employ the one-dimensional approach with the partition shape of $\mathbb{d}_{(\text{par})} = (4)$. The experimental configuration in obtaining these results will be described in Sec. 5.

Table 2: Word Error Rates (WERs) with the RNN-T model shown in Fig. 3a using the scaling suggested by (5) and $\frac{1}{1-p}$. The dropout rate is 0.2 and the partition shape for the 1-dimensional *macro-block dropout* is $\mathbb{d}_{(\text{par})} = (4)$.

Test Set	Baseline Dropout	1-D Macro-Block Dropout	
		Scaling using (5)	Scaling using $\frac{1}{1-p}$
`test-clean`	**3.95** %	**3.78** %	4.04 %
`test-other`	**12.23** %	**11.48** %	11.50 %
Average	**8.09** %	**7.63** %	7.77 %

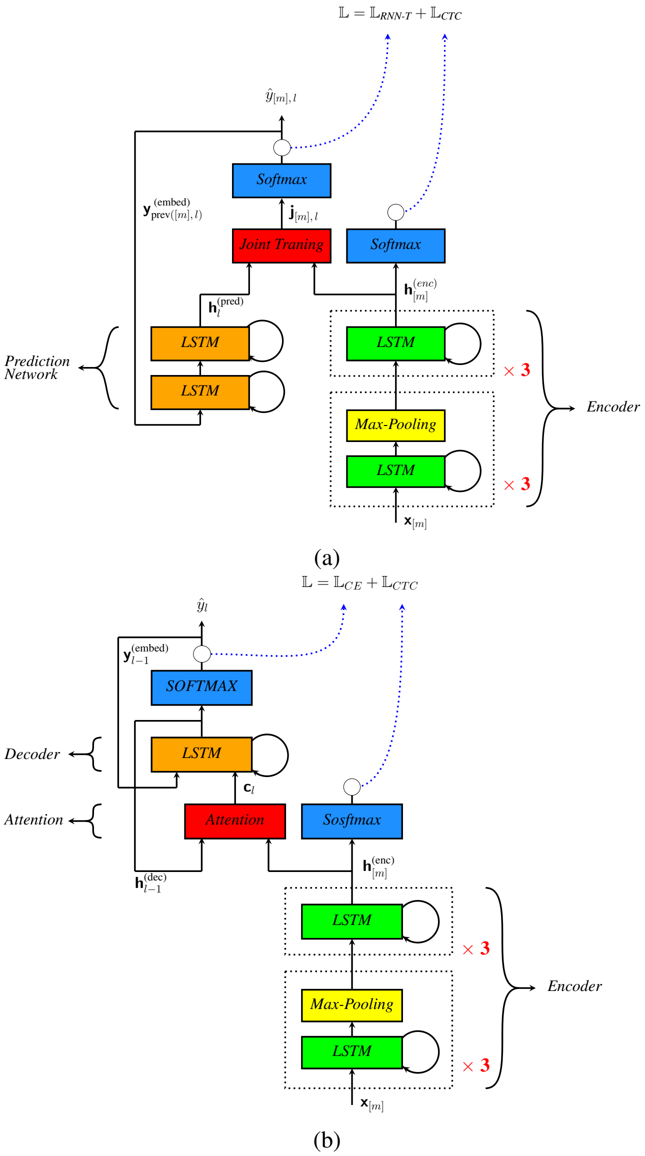
4. SPEECH RECOGNITION MODEL

Speech recognition is a task of finding the *sequence-to-sequence* mapping from an input sequence of acoustic features to a output sequence of labels [22]. Let us denote the input and output sequences by $\mathbf{x}_{[0:M]}$ and $y_{0:L}$ as shown below:

$$\mathbf{x}_{[0:M]} = \left[\mathbf{x}_{[0]}, \mathbf{x}_{[1]}, \mathbf{x}_{[2]}, \cdots, \mathbf{x}_{[M-1]}\right], \tag{7a}$$

$$y_{0:L} = \left[y_0, y_1, y_2, \cdots, y_{L-1}\right], \tag{7b}$$

where M and L are the lengths of the input acoustic feature sequence and the output label sequence, respectively. The sequence notation adopted in this paper including (7) follows the `Python` array slice notation. In this paper, by convention, we use a *bracket* to



$\mathbb{L} = \mathbb{L}_{RNN\text{-}T} + \mathbb{L}_{CTC}$
$\hat{y}_{[m],l}$
$\mathbf{y}^{(\text{embed})}_{\text{prev}([m],l)}$
Softmax
$\mathbf{j}_{[m],l}$
Joint Traning
Softmax
$\mathbf{h}^{(enc)}_{[m]}$
$\mathbf{h}^{(\text{pred})}_{l}$
Prediction Network
LSTM
LSTM
LSTM
× 3
Max-Pooling
LSTM
× 3
Encoder
$\mathbf{x}_{[m]}$
(a)
$\mathbb{L} = \mathbb{L}_{CE} + \mathbb{L}_{CTC}$
$\hat{y}_l$
$\mathbf{y}^{(\text{embed})}_{l-1}$
SOFTMAX
Decoder
LSTM
$\mathbf{c}_l$
Attention
Attention
Sosftmax
$\mathbf{h}^{(\text{enc})}_{[m]}$
$\mathbf{h}^{(\text{dec})}_{l-1}$
LSTM
× 3
Max-Pooling
LSTM
× 3
Encoder
$\mathbf{x}_{[m]}$
(b)

Fig. 3: The structures of (3a) the Recurrent Neural Network-Transducer (RNN-T) speech recognizer and (3b) the Attention-based Encoder Decoder (AED) speech recognizer used in this work.

represent a periodically sampled sequence such as the acoustic feature, and use a *subscript* to represent a non-periodic sequence such as the output label.

4.1. RNN-T and AED models

For speech recognition experiments, we employed an RNN-T speech recognizer and an Attention-based Encoder Decoder (AED), whose structures are shown in Fig. 3a and 3b, respectively. Our speech recognition system is built *in-house* using `Keras` models [23] implemented for `Tensorflow` 2.3 [24]. The RNN-T structures have three major components: an encoder (also known as a transcription network), a prediction network, and a joint network. In our implementation, the encoder consists of six layers of bi-directional LSTMs interleaved with 2:1 max-pooling layers [25] in the bottom three layers. Thus, the overall temporal sub-sampling factor is 8:1 because of these three 2:1 max-pooling layers. The prediction network consists of two layers of uni-directional LSTMs. The unit size of all these LSTM layers is 1024. *Macro-block dropout* is applied to the output of each LSTM layer in the encoder except the top-most layer and to the output of each LSTM layer in the prediction network. $y^{(\text{embed})}_{l-1}[m]$ is a linear embedding vector with the dimension of 621 obtained from the output and fed back into the prediction network. The AED model has three components as shown in Fig. 3b: an encoder, a decoder, and an attention block. In our implementation, the encoder of the AED model is identical to the encoder structure of the RNN-T model explained above. We use a single layer of uni-directional LSTM whose unit size is 1024.

The loss employed for training the RNN-T model is a combination of the Connectionist Temporal Classification (CTC) loss [26] applied to the *encoder* output and the RNN-T loss [5] applied to the full network that is represented by the following:

$$\mathbb{L}_{\text{CTC-RNN-T}} = \mathbb{L}_{\text{CTC}} + \mathbb{L}_{\text{RNN-T}}, \tag{8}$$

which is inspired by the work in [27]. We refer this loss in (8) to as the *joint CTC-RNN-T loss*. For the AED model, we employ the joint *CTC-CE loss*, which is given by:

$$\mathbb{L}_{\text{CTC-CE}} = \mathbb{L}_{\text{CTC}} + \mathbb{L}_{\text{CE}}, \tag{9}$$

which has been also used in our previous works [28, 22]. For a better stability during the training, we use the gradient clipping by global norm [29], which is implemented in Tensorflow [24] as the `tf.clip_by_global_norm` API.

4.2. Improved shallow-fusion with a language model

End-to-end speech recognition models in Sec. 4.1 are trained using only paired speech-text data. Compared to traditional AM-LM approaches where an LM is often trained using a much larger text corpus possibly containing billions of sentences [30], the end-to-end speech recognition model sees much limited number of word sequences during the training phase. To get further performance improvement, various techniques of incorporating external language models such as *shallow-fusion* [31], *deep-fusion* [32], and *cold-fusion* [33] have been proposed. Among them, in spite of its simplicity, shallow-fusion seems to be more effective than other approaches [34]. In shallow-fusion, the log probability from the end-to-end speech recognition model is linearly interpolated with the probability from the language model as follows:

$$\log p_{\text{sf}}\left(y_l \middle| \mathbf{x}_{[0:m]}\right) = \log p\left(y_l \middle| \mathbf{x}_{[0:m]}, \hat{y}_{0:l}\right) + \lambda \log p_{\text{lm}}\left(y_l \middle| \hat{y}_{0:l}\right), \tag{10}$$

where $p_{\text{lm}}\left(y_l \middle| \hat{y}_{0:l}\right)$ is the probability of predicting the label y_l from the LM, and the $p\left(y_l \middle| \mathbf{x}_{[0:m]}\right)$ is the posterior probability obtained from the end-to-end speech recognition model.

In [11], we proposed an improved version of this shallow fusion. In this approach, we introduce another term to subtract the log prior probability of each label obtained from the training database for the speech recognition model. The motivation is that this prior probability might give too much bias in predicting the next label. This improved shallow fusion is given by the following equation:

$$\log p_{\text{sf}}\left(y_l \middle| \mathbf{x}_{[0:m]}\right) = \log p\left(y_l \middle| \mathbf{x}_{[0:m]}, \hat{y}_{0:l}\right) - \lambda_p \log p_{\text{prior}}\left(y_l\right) + \lambda_{\text{lm}} \log p_{\text{lm}}\left(y_l \middle| \hat{y}_{0:l}\right), \tag{11}$$

Table 3: Word Error Rates (WERs) with the RNN-T model shown in Fig. 3a using the baseline dropout and the one-dimensional *maro-block dropout* approaches. In these experiments, the dropout rate of 0.2 is used since the best WER in each case is obtained at this rate.

Test Set	Baseline Dropout	One-Dimensional Macro-Block Dropout			
		Number of Blocks			
		3	4	5	10
test-clean	**3.95** %	4.11 %	**3.78** %	3.88 %	3.94 %
test-other	**12.23** %	11.57 %	**11.48** %	11.52 %	11.50 %
Average	**8.09** %	7.84 %	**7.63** %	7.70 %	7.72 %

Table 4: Word Error Rates (WERs) with the Attention-based Encoder Decoder (AED) model shown in Fig. 3b using the baseline dropout and the one-dimensional *maro-block dropout* approaches. In these experiments, the dropout rate of 0.2 is used since the best WER in each case is obtained at this rate.

Test Set	Baseline Dropout	One-Dimensional Macro-Block Dropout			
		Number of Blocks			
		3	4	5	10
test-clean	**3.67** %	3.66 %	**3.51** %	3.54 %	3.61 %
test-other	**11.62** %	11.20 %	**10.94** %	10.98 %	11.07 %
Average	**7.65** %	7.43 %	**7.23** %	7.26 %	7.34 %

where we have an additional term $\lambda_p \log p_{\text{prior}}(y_l)$ for subtracting the prior bias that the model has learned from the speech recognition training corpus. In our experiments, we use λ_p of 0.002 and λ_{lm} of 0.48 respectively.

5. EXPERIMENTAL RESULTS

Table 5: Word Error Rates (WERs) with the Attention-based Encoder Decoder model shown in Fig. 3b with an improved shallow fusion in (11) with a Transformer LM [35, 36].

Test Set	Baseline Dropout	Macro-Block Dropout
test-clean	**2.44** %	**2.37** %
test-other	**7.87** %	**7.42** %
Average	**5.16** %	**4.90** %

In this section, we explain experimental results using the *macro-block dropout* approach with the RNN-T and the AED model described in Sec. 4. For training, we used the entire 960 hours LibriSpeech [37] training set consisting of 281,241 utterances. For evaluation, we used the official 5.4 hours test-clean and 5.1 hours test-other sets consisting of 2,620 and 2,939 utterances respectively. The pre-training stage has some similarities to our previous work in [38]. In this pre-training stage, the number of LSTM layers in the encoder increased at every 10,000-steps starting from two LSTM layers up to six layers. We use an Adam optimizer [39] with the initial learning rate of 0.0003, which is maintained for the entire pre-training state and one full epoch after finishing the pre-training stage. After this step, this learning rate decreases exponentially with a decay rate of 0.5 for each epoch. $\mathbf{x}[m]$ and $\mathbf{y}_l$ are the input *power-mel filterbank* feature of size 40 and the output label, respectively. m is the input frame index and l is the decoder output step index. We use the *power-mel filterbank* feature instead of the more commonly used *log-mel filterbank* feature based on our previous results [38, 11]. For better regularization in training, we apply the *SpecAugment* as a data-augmentation technique in all the experiments in the paper [10].

In our experiments, we observe that with both the conventional and the *macro-block* dropout approaches, the best Word Error Rates (WERs) are obtained when the dropout rate p is close to 0.2. Table 3 summarizes the experimental results with conventional dropout and *macro-block* dropout approaches using the RNN-T model. In the case of the *macro-block* dropout approach, we conducted experiments with four different partition sizes with the one-dimensional masking pattern as shown in Fig. 2b. For the LibriSpeech test-other set, the best WER is obtained when the number of blocks is four, even though there is not much variation in WERs depending on the number of blocks. As shown in this table, *macro-block* dropout algorithm has shown relatively 4.30 % and 6.13 % Word Error Rate (WER) improvements over the conventional dropout approach on LibriSpeech test-clean and test-other. Table 4 shows the results with conventional dropout and *macro-block* approaches using the AED model. We observe that the performance improvement using the AED model in Table 4 is similar to that using the RNN-T model in Table 3. We obtain 4.36 % and 5.85 % relative WER improvements on the same LibriSpeech test-clean and test-other sets.

Finally, we apply the improved shallow fusion in 11 to further improve the performance. Table 5 shows WERs obtained with the AED model using the improved shallow fusion in (11) with a Transformer LM [35, 36]. As shown in this Table, *macro-block dropout* shows 2.86 % and 5.72 % relative WER improvement on the *LibriSpeech* `test-clean` and `test-other` sets.

6. CONCLUSIONS

In this paper, we described a new regularization algorithm referred to as *macro-block dropout*. In this approach, rather than applying dropout to each element, we apply random mask to a bigger chunk referred to as a *macro-block*. The scaling after masking is also improved for better performance. In our experiments using a Recurrent Neural Network- Transducer (RNN-T), this simple algorithm has shown relatively 4.30 % and 6.13 % Word Error Rate (WER) improvements over the conventional dropout approach on LibriSpeech `test-clean` and `test-other`, which is even significantly larger than the relative WER improvements of the conventional dropout approach over the no-dropout case. In experiments using an Attention-based Encoder Decoder (AED) model, this *macro-block dropout* approach shows relatively 4.36 % and 5.85 % Word Error Rate (WER) improvements over the conventional dropout approach on the same LibriSpeech `test-clean` and `test-other` sets. The *Keras* layer implementation of this algorithm will be released as open-source.

7. REFERENCES

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, Nov. 2012.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, no. 9, pp. 1735–1780, Nov. 1997.

[3] C. Kim, A. Misra, K. Chin, T. Hughes, A. Narayanan, T. N. Sainath, and M. Bacchiani, "Generation of large-scale simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in google home," in *Proc. Interspeech 2017*, 2017, pp. 379–383. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1510

[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 577–585. [Online]. Available: http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition.pdf

[5] A. Graves, A. rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6645–6649.

[6] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4774–4778.

[7] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 66–75. [Online]. Available: https://www.aclweb.org/anthology/P18-1007

[8] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 4839–4843.

[9] C. Kim, K. Kim, and S. Indurthi, "Small energy masking for improved neural network training for end-to-end speech recognition," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020, pp. 7684–7688.

[10] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[11] C. Kim, M. Shin, A. Garg, and D. Gowda, "Improved vocal tract length perturbation for a state-of-the-art end-to-end speech recognition system," in *INTERSPEECH-2019*, Graz, Austria, Sept. 2019, pp. 739–743. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-3227

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[13] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1058–1066. [Online]. Available: http://proceedings.mlr.press/v28/wan13.html

[14] G. Larsson, M. Maire, and G. Shakhnarovich, "Fractalnet: Ultra-deep neural networks without residuals," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=S1VaB4cex

[15] X. Gastaldi, "Shake-shake regularization of 3-branch residual networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=HkO-PCmYl

[16] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "Dropblock: A regularization method for convolutional networks," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 10 727–10 737. [Online]. Available: http://papers.nips.cc/paper/8271-dropblock-a-regularization-method-for-convolutional-networks.pdf

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[18] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8609–8613.

[19] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise, “Shakedrop regularization for deep residual learning,” *IEEE Access*, vol. 7, pp. 186 126–186 136, 2019.

[20] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 1019–1027. [Online]. Available: http://papers.nips.cc/paper/6241-a-theoretically-grounded-application-of-dropout-in-recurrent-neural-networks.pdf

[21] C. Kim, D. Gowda, D. Lee, J. Kim, A. Kumar, S. Kim, A. Garg, and C. Han, “A review of on-device fully neural end-to-end automatic speech recognition algorithms,” in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2020.

[22] ——, “A review of on-device fully neural end-to-end automatic speech recognition algorithms,” in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2020.

[23] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[24] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[25] M. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun, “Unsupervised learning of invariant feature hierarchies with applications to object recognition,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML ’06. New York, NY, USA: ACM, 2006, pp. 369–376. [Online]. Available: http://doi.acm.org/10.1145/1143844.1143891

[27] 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)*, 2017, pp. 4835–4839.

[28] C. Kim, A. Garg, D. Gowda, S. Mun, and C. Han, “Streaming end-to-end speech recognition with jointly trained neural feature enhancement,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6773–6777.

[29] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ser. ICML’13. JMLR.org, 2013, pp. III–1310–III–1318. [Online]. Available: http://dl.acm.org/citation.cfm?id=3042817.3043083

[30] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 5824–5828.

[31] J. Chorowski and N. Jaitly, “Towards better decoding and language model integration in sequence to sequence models,” in *Proc. Interspeech 2017*, 2017, pp. 523–527. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-343

[32] Ç. Gülçehre, O. Firat, K. Xu, K. Cho, L. Barrault, H. Lin, F. Bougares, H. Schwenk, and Y. Bengio, “On using monolingual corpora in neural machine translation,” *CoRR*, vol. abs/1503.03535, 2015. [Online]. Available: http://arxiv.org/abs/1503.03535

[33] A. Sriram, H. Jun, S. Satheesh, and A. Coates, “Cold fusion: Training seq2seq models together with language models,” 2017.

[34] S. Toshniwal, A. Kannan, C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, “A comparison of techniques for language model integration in encoder-decoder speech recognition,” in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 369–375.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[36] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, “RWTH ASR systems for librispeech: Hybrid vs attention - w/o data augmentation,” *CoRR*, vol. abs/1905.03072, 2019. [Online]. Available: http://arxiv.org/abs/1905.03072

[37] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[38] C. Kim, S. Kim, K. Kim, M. Kumar, J. Kim, K. Lee, C. Han, A. Garg, E. Kim, M. Shin, S. Singh, L. Heck, and D. Gowda, “End-to-end training of a large vocabulary end-to-end speech recognition system,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec. 2019, pp. 562–569.

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980