**Attention Enhanced Citrinet for Speech Recognition**

Xianchao Wu
NVIDIA
xianchaow@nvidia.com

**Abstract**

Citrinet is an end-to-end convolutional Connectionist Temporal Classification (CTC) based automatic speech recognition (ASR) model. To capture local and global contextual information, 1D time-channel separable convolutions combined with sub-word encoding and squeeze-and-excitation (SE) are used in Citrinet, making the whole architecture to be as deep as including 23 blocks with 235 convolution layers and 46 linear layers. This pure convolutional and deep architecture makes Citrinet relatively slow at convergence. In this paper, we propose to introduce multi-head attentions together with feed-forward networks in the convolution module in Citrinet blocks while keeping the SE module and residual module unchanged. For speeding up, we remove 8 convolution layers in each attention-enhanced Citrinet block and reduce 23 blocks to 13. Experiments on the Japanese CSJ-500h and Magic-1600h dataset show that the attention-enhanced Citrinet with less layers and blocks and converges faster with lower character error rates than (1) Citrinet with 80% training time and (2) Conformer with 40% training time and 29.8% model size.

**Index Terms:** speech recognition, human-computer interaction, computational paralinguistics

1. **Introduction**

Convolutions with predefined kernel sizes capturing local contexts have been successfully utilized in end-to-end ASR systems. JasperNet [1], QuartzNet [2], ContextNet [3], and Citrinet [4] follow this direction and use pure convolution networks to capture the interactive context information of input wave spectrogram. Among these models, Citrinet (Figure 1) has achieved top-level accuracy. Citrinet’s encoder combines 1D time-channel separable convolutions following QuartzNet [2] and the SE mechanism [5] which has been successfully used in ContextNet [3] for global context representation.

Alternatively, ASR systems leveraged on Transformers [6] and their variants have achieved impressively low word/character error rates (WERs/CERs) in numerous languages during recent years [7, 8, 9, 10, 11]. The Transformer architecture includes multi-head self-attention (MHSA) and cross-attention layers that can represent long-distance interactions inside and between sound-text sequences.

There are works such as Conformer [8] which introduced convolutions into Transformer architecture and achieved impressive WERs/CERs in numerous languages. In Conformer, two macaron-like [12] feed-forward networks (FFN) with half-step residual connections sandwiches the MHSA and convolution modules followed by a post layer normalization [13].

In this paper, we consider a reverse direction of introducing attention mechanisms into convolution-leading architecture. In particular, we introduce self-attention and cross-attention in a state-of-the-art deep convolutional CTC model, Citrinet, with as minimum modifications as possible. Our work is motivated by the following facts and considerations. First, in order to capture long-range contextual representation, there are 235 convolution layers and 46 linear layers in a typical “thin and long” Citrinet model. As will be described in detail in Section 2, there are 10 convolutional functions stacked in one Jasper block used in Citrinet without internal residual connection. These make the training convergence to be relatively slow and the gradient back-propagation is relatively more difficult without shortcut-style residual connections [14]. Second, kernel sizes from 11 to 39 and further 41 are hard-coded in Citrinet to capture as long range information as possible. This hard setting is less friendly to long and short speech waves: a kernel size of 41 is still not enough for a long wave yet it can be a waste for short waves. Third, current Citrinet is a pure CTC model with a non-autoregressive decoder: one convolutional layer projects from the encoder’s output to the vocabulary size. We are interested in boosting Citrinet with cross-attention enhanced decoder architecture.

In our attention-enhanced Citrinet, we insert a FFN module and a MHSA module in the “1D conv module” of one Jasper block and then reduce the repeating time from 5 to 1 to remove 80% of convolution layers and select 13 from 23 blocks to obtain a fat and short pipeline. We also add a bidirectional Transformer [6] decoder to ensure Citrinet’s training in a non-autoregressive way using a linear combination of CTC loss and attention losses, and inferring in the same non-autoregressive way. We perform our experiments using the 500-hour Japanese CSJ dataset and the 1600-hour Japanese Magic dataset.

---

**Figure 1:** The end-to-end architecture of Citrinet and its major building block - “1D time-channel separable convolution and SE-attached” Jasper Block.
2. Attention Enhanced Citrinet

2.1. Citrinet

Figure 1 depicts the “wave to textual sequence” pipeline of Citrinet and its major building block. The preprocessing is to extract the 80-dimension FBanks and then perform spectrogram augmentation [15] on time/frequency dimensions. One (updated) Jasper block [1, 2], in the right-hand-side of the figure, contains three modules: (1) one 1D time-channel separable convolution module (Figure 2), (2) one SE module [5] and in parallel (3) one residual (Res) connection module (Figure 3). This Jasper block can be expressed by the following equations:

\[ \hat{X} = 1D_{\text{Conv}}(X) \]  \hspace{1cm} (1)

\[ X' = \text{Res}(X) + \hat{X} \times SE(\hat{X}). \]  \hspace{1cm} (2)

The “1D conv module” repeatedly extracts time-channel contextual representations guided by increased kernel sizes. There are two linear layers in the SE module that project from \( C \) to \( C/8 \) and back to \( C \) to capture the whole sequence contextual information of each channel. The contextual information is then projected into the \([0, 1]\) range by a sigmoid function to act as gates controlling the traffic to the following layers. Thus, the local and global information are extracted by 1D conv and SE modules, respectively. The Res module takes the original tensor \( X \) as input and performs a simple \( 1 \times 1 \) convolution followed by a batch normalization. The stride for this \( 1 \times 1 \) convolution can take values of 1 or 2 to align with the distilling of the three mega blocks to be described hereafter.

The encoder contains 23 Jasper blocks which are assigned to 5 block sets: one prolog block, three mega blocks and finally one epilog block. Prolog block and epilog block both adapt the Jasper block architecture except that the residual module (“Res Module” in the Figure) is not included. Using the “1D Conv Module”, prolog block’s major task is to transform the input 80-dimension FBanks into \( C \) (e.g., 384, 1024) channels and epilog block’s major task is to capture global information again with kernel size 41 and then transform the input \( C \) channels into the dimension (e.g., 640) similar to the decoder’s input.

There are 6, 7, and 8 Jasper blocks in the three mega blocks, respectively. In each first Jasper block, a stride of 2 is used and consequently the “time-dimension” length is halved. Generally, this can be recognized as a distilling process in the encoder: output length is \( 1/2^3 \) of input length. In order to capture global context information, the kernel sizes of the mega blocks increase linearly from 11 to 39. In particular, the first mega block’s kernel sizes range over 11, 13, ..., to 21; the second mega block takes values from 13, 15, ..., to 25; and the third mega block starts from kernel size of 25 and ends at 39.

The decoder is a one-layer convolutional network that projects from the encoder’s output to the vocabulary size. Finally, CTC loss [16] is used during training the architecture.

2.2. Enhance 1D Time-channel Convolution by Attention

Figure 2 shows the original 1D time-channel separable convolution module and our modifications in red font after arrows. This depth-wise separable network includes (1) a depth-wise CNN with kernel size from 11 to 39 that tries to capture relatively a long range of context and (2) a pointwise CNN that acts like a linear projection with kernel size of 1. Then, batch normalization, ReLU activation and dropout are performed. In the original Jasper block, these repeat 5 times resulting in 10 convolutional functions, 5 batch normalization and 4 ReLU/dropout functions (the final repeat does not include ReLU/dropout). These 10 convolutional layers are stacked together without any residual connections inside them. As shown in Figure 1, there is only one residual module outside the 1D conv module. These plain nets that simply stack CNN layers (without shortcut-style residual connections in between) exhibit higher training error when the depth increases as illustrated in ResNet [14].

Motivated by this observation and the successful combination of CNNs with attentions in Conformer [8], we modify this 1D conv module by (1) setting repeat time from 5 to 1, (2) appending a FFN module and a MHSA module right before the first convolutional layer, (3) changing the batch nor-
malization into layer normalization [13] which has been successfully used in Transformer [6] and (4) replacing ReLU by the Swish [17] activation function which is a self-gated function \( f(x) = x \cdot \text{sigmoid}(x) \) and has been successfully used in Conformer [8].

We reduce 8 convolution functions in one Jasper block. The appending of the FFN and MHSA modules bring 6 linear projection layers, 2 layer normalization and 2 residual connections, making the resulting Citrinet architecture to be fatter and shorter. We count on these FFN+MHSA modules for capturing global interactive information among elements of the input sequence. This works together with the SE module to perform self-attending and self-gating of global information.

2.3. Enhance Decoder by Attention

We append a 3-block bidirectional Transformer decoder which encodes the textual sequences in left-to-right (l2r) and right-to-left (r2l) directions and predicts the whole target sequence by one forward step. The input and reference output to the \( l2r \) decoder are \( \text{"s, } y_{1}, \ldots, y_{N} \) \( \text{"y}_{1}, \ldots, y_{N}, /s \) which are taken from a reference label sequence \( y_{1}, \ldots, y_{N} \), respectively. Here, \( \text{s} \) and \( /s \) stand for start/end of sequence symbols, respectively. The \( r2l \) decoder takes \( "s, y_{N}, \ldots, y_{1} \) as input and predicts \( y_{1}, \ldots, y_{N}, /s \).

The objective is to minimize \( \mathcal{L} \), a linear combination of the CTC loss [16] and attention losses (ATT) computed by point-wise KL-divergence [18]:

\[
\mathcal{L} = \lambda_1 \cdot \text{CTC}_{\text{loss}} + (1 - \lambda_1) \ast \{\lambda_2 \ast \text{ATT}_{l2r} + (1 - \lambda_2) \ast \text{ATT}_{r2l}\}
\]

(3)

When we set \( \lambda_1 = 1 \), it degenerates to the original Citrinet’s loss function. In our attention-enhanced Citrinet variants, we set \( \lambda_1 = 0.3 \) and \( \lambda_2 = 0.7 \). Label smoothing with \( \delta = 0.1 \) is applied to the attention objective so that the references are discounted by \( (1-\delta) \).

3. Experiments

3.1. Setup

Our experiments are performed under an open-source platform, NVIDIA NeMo\(^v\) which is a conversational AI toolkit built for researchers working on ASR, natural language processing, and text-to-speech synthesis with rich pretrained models and complete implementation of Citrinet. We investigate Citrinet and attention-enhanced Citrinet’s performances under channel sizes from 256 to 1,048. We release the code and pretrained models through NeMo. During training, we select the Novograd optimizer \([19] \) (\( \beta = [0.8, 0.25] \)) with a start learning rate of 0.05. The cosine annealing \([20] \) learning rate scheduler with \( \beta \)-factor is applied to the attention objective so that the references are discounted by \( (1-\delta) \).

3.2. Data

We evaluate attention-enhanced Citrinet variants on the Japanese “Corpus of Spontaneous Japanese” (CSJ) dataset\(^3\) and the Japanese Magic dataset\(^4\), which respectively consist of 500 and 1,600 hours of labeled waves. The CSJ dataset contains research presentations that are recorded from conference rooms. The Magic dataset is relatively noisier of spoken Japanese with informal Japanese grammar. Both datasets are Tokyo-region pronunciations. We use sentencepiece-bpe \([22] \) to segment the text sequences and select a vocabulary size of 4,096. Both datasets have 3,000 around single characters and 92% characters are overlapping. During data preparation, we generate 80-dimension FBank feature vectors with a 25ms window, a 10ms frame stride and dither=1.0. SpecAugment \([15] \) is adapted with 2 frequency masks \((F=10)\) and 2 time masks \((T=50)\). Global Cepstral Mean and Variance Normalization \([23] \) technique is employed to normalize the 80-dimension FBank vectors.

3.3. Major Results on the CSJ Test Sets

Table 1 lists the CERs (%) of 5 Citrinet and attention-enhanced Citrinet variants and 3 baselines. The first baseline is a Transformer with enhanced information from speakers and speech \([24] \). Parameters were shared among layers of encoder and decoder blocks in Transformer. Speaker information (gender, age, education-level, speaker ID) and speech utterance properties (duration of short and long, topic of the lecture) were used to augment the training data. The second baseline is Espnet’s implementation of deep VGG-BLSTM \([25] \) with CTC joint decoding and LM rescoring\(^5\). Note that only this baseline uses an external language model among all the 13 models. The third baseline is Conformer-Large \([8] \) with 135.1M parameters.

Since convolutions play a leading position in Citrinet, we take channel sizes of 256, 384, 512, 768, and 1,024 to investigate if enlarging the model size yields lower CERs. From Table

<table>
<thead>
<tr>
<th>Model</th>
<th>1.73h test1</th>
<th>1.82h test2</th>
<th>1.23h test3</th>
<th>Model size (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer [24]</td>
<td>7.6</td>
<td>6.1</td>
<td>6.3</td>
<td>36</td>
</tr>
<tr>
<td>Espnet+LM</td>
<td>6.5</td>
<td>4.6</td>
<td>5.1</td>
<td>NA</td>
</tr>
<tr>
<td>Conformer [8]</td>
<td>6.97</td>
<td>4.65</td>
<td>5.29</td>
<td>135.1</td>
</tr>
<tr>
<td>C-256</td>
<td>8.14</td>
<td>5.79</td>
<td>6.54</td>
<td>12.2</td>
</tr>
<tr>
<td>C-384</td>
<td>7.28</td>
<td>4.81</td>
<td>5.44</td>
<td>22.7</td>
</tr>
<tr>
<td>C-512</td>
<td>7.54</td>
<td>5.45</td>
<td>6.01</td>
<td>38.0</td>
</tr>
<tr>
<td>C-768</td>
<td>7.37</td>
<td>5.33</td>
<td>5.71</td>
<td>85.0</td>
</tr>
<tr>
<td>C-1024</td>
<td>7.58</td>
<td>5.76</td>
<td>5.87</td>
<td>141.6</td>
</tr>
<tr>
<td>Att-C-256</td>
<td>7.10</td>
<td>4.75</td>
<td>5.79</td>
<td>19.9</td>
</tr>
<tr>
<td>Att-C-384</td>
<td>6.27</td>
<td>4.16</td>
<td>5.01</td>
<td>40.3</td>
</tr>
<tr>
<td>Att-C-512</td>
<td>6.99</td>
<td>4.94</td>
<td>5.46</td>
<td>68.6</td>
</tr>
<tr>
<td>Att-C-768</td>
<td>7.23</td>
<td>4.87</td>
<td>5.18</td>
<td>123.5</td>
</tr>
<tr>
<td>Att-C-1024</td>
<td>7.23</td>
<td>5.49</td>
<td>5.55</td>
<td>201.4</td>
</tr>
</tbody>
</table>

\(^v\)https://github.com/NVIDIA/NeMo
\(^3\)https://ccl.ninjal.ac.jp/csj/en/
\(^5\)https://github.com/espnet/espnet/blob/master/egs/csj/asr1/RESULTS.md
Table 2: CERs (%) of 3 baselines and 3 Citrinet (C) and attention-enhanced Citrinet (Att-C) variants with different channel sizes. All models are trained under the combined CSJ-500h + Magic-1600h dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>test1</th>
<th>test2</th>
<th>test3</th>
<th>Magic-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conformer [8]</td>
<td>11.22</td>
<td>6.35</td>
<td>8.78</td>
<td>11.73</td>
</tr>
<tr>
<td>C-256</td>
<td>13.68</td>
<td>7.78</td>
<td>7.29</td>
<td>12.65</td>
</tr>
<tr>
<td>C-384</td>
<td>8.51</td>
<td>5.80</td>
<td>8.36</td>
<td>10.36</td>
</tr>
<tr>
<td>C-512</td>
<td>12.39</td>
<td>7.25</td>
<td>7.72</td>
<td>13.98</td>
</tr>
<tr>
<td>Att-C-256</td>
<td>12.02</td>
<td>7.22</td>
<td>7.54</td>
<td>10.80</td>
</tr>
<tr>
<td>Att-C-384</td>
<td>7.94</td>
<td>5.62</td>
<td>6.92</td>
<td>9.13</td>
</tr>
<tr>
<td>Att-C-512</td>
<td>11.53</td>
<td>7.21</td>
<td>7.60</td>
<td>11.98</td>
</tr>
</tbody>
</table>

Table 3: CERs (%) of variants when deleting individual modules on the CSJ test sets.

<table>
<thead>
<tr>
<th>Model</th>
<th>test1</th>
<th>test2</th>
<th>test3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-384</td>
<td>7.28</td>
<td>4.81</td>
<td>5.44</td>
</tr>
<tr>
<td>Att-C-384</td>
<td>6.27</td>
<td>4.16</td>
<td>5.01</td>
</tr>
<tr>
<td>without bi-decoder</td>
<td>7.08</td>
<td>4.51</td>
<td>5.24</td>
</tr>
<tr>
<td>without FFN</td>
<td>6.95</td>
<td>4.65</td>
<td>5.25</td>
</tr>
<tr>
<td>layer-norm → batch-norm</td>
<td>6.58</td>
<td>4.46</td>
<td>5.21</td>
</tr>
<tr>
<td>Swish → ReLU</td>
<td>6.35</td>
<td>4.29</td>
<td>5.23</td>
</tr>
</tbody>
</table>

Table 2 lists the CERs of three Citrinet and attention-enhanced Citrinet variants and the Conformer baseline with the same configuration as Table 1. We choose channel sizes 256, 384, and 512. From Table 2, we observe that C-384 again performs the best and larger channel sizes do not bring reduction of CERs: C-256 and C-512’s performances are comparable. Similar tendency happens in attention-enhanced Citrinet.

We have the following observations. First, CSJ’s three test sets’ CERs become worse when different domain dataset is merged into the relatively clean CSJ dataset. Second, the original Citrinet performs better than the Conformer, with 1.21% (relatively 12.7%) CER decreasing on average of the four test sets. This reflects that a deeper convolutional model trained using large-scale datasets is still robust. Finally, attention-enhanced Citrinet outperforms Citrinet of all the four test sets with a 0.91% (relatively 10.9%) CER reduction on average.

3.5. Impact of Individual Modules

We perform an ablation study of the individual ideas in the attention-enhanced Citrinet. Table 3 shows the CER results when independently removing bi-decoder, dropping FFN modules, replacing layer normalization with batch normalization and changing Swish activation function back to ReLU. Bi-decoder and FFN are relatively important among the four ideas.

3.6. Ablation Study of Jasper Block Numbers

We finally investigate the number of total Jasper blocks with a value set of {8, 9, 10, 11, 12, 13, 15, 17, 20}. We averagely prune ending Jasper blocks in the 3 mega block set (Figure 2). The variants achieve stable comparable CERs with a standard deviation of 0.3% except 9 and 12 with +2% CERs. We thus select the first stable block number of 13.

4. Conclusions

In this paper, we introduce attention mechanisms into a convolution-leading architecture, Citrinet, for end-to-end non-autoregressive Japanese ASR. We keep Citrinet’s architecture of using 1D time-channel separable convolutions, SE and residual modules and append FFN and MHSA before each 1D time-channel convolution. We reduce 8 convolution layers in each attention-enhanced Citrinet block and only utilize 13 such blocks. Experiments on the Japanese CSJ-500h and Magic-1600h datasets show that the attention-Citrinet requires less layers of blocks and converges faster with lower character error rates than Citrinet with 80% training time and Conformer [8] with 40% training time and 29.8% model size. Future work includes using linear attentions [26, 27] and adaptive span attentions [28] for building more efficient Citrinet.
5. References


