Citrinet: Closing the Gap between Non-Autoregressive and Autoregressive End-to-End Models for Automatic Speech Recognition

Somshubra Majumdar, Jagadeesh Balam, Oleksii Hrinchuk, Vitaly Lavrukhin, Vahid Noroozi, Boris Ginsburg

NVIDIA, USA

{smajumdar,jbalam,ohrinchuk,vlavrukhin,vnoroozi,bginsburg}@nvidia.com

Abstract

We propose Citrinet - a new end-to-end convolutional Connectionist Temporal Classification (CTC) based automatic speech recognition (ASR) model. Citrinet is deep residual neural model which uses 1D time-channel separable convolutions combined with sub-word encoding and squeeze-and-excitation. The resulting architecture significantly reduces the gap between non-autoregressive and sequence-to-sequence and transducer models. We evaluate Citrinet on LibriSpeech, TED-LIUM 2, AISHELL-1 and Multilingual LibriSpeech (MLS) English speech datasets. Citrinet accuracy on these datasets is close to the best autoregressive Transducer models.

1. Introduction

End-to-end neural ASR models can be roughly classified into three groups based on the network architecture and loss type:[1]

1. Connectionist Temporal Classification (CTC) models [3]
2. Sequence-to-sequence (Seq2Seq) models with attention, e.g. Listen-Attend-Spell [4, 5, 6] or Transformer with Seq2Seq loss [7]
3. RNN-Transducers (RNN-T) [8]

Seq2Seq and RNN-T models are autoregressive and sequential decoding makes them slower to train and evaluate than non-autoregressive models. CTC models have the benefit of being more stable and easier to train than autoregressive models, but the latest Seq2Seq models and RNN-Transducers significantly outperform CTC models, see Table 1. The difference in the accuracy between CTC and autoregressive models is usually explained by the claim that “because of the strong conditional independence assumption...CTC does not implicitly learn a language model over the data (unlike the attention-based encoder-decoder architectures). It is therefore essential when using CTC to interpolate a language model”[10]. We show that CTC models can overcome these limitations by using recent advances in NN architectures.

In this paper, we describe Citrinet - a deep convolutional CTC model. The Citrinet encoder combines 1D time-channel separable convolutions from QuartzNet [9] with the squeeze-and-excite (SE) mechanism [17] from the ContextNet [14]. Citrinet significantly closes the gap between CTC and the best Seq2Seq and Transducers models. Without any external LM, the Citrinet-1024 model reaches Word Error Rate (WER) 6.22% on LibriSpeech [18] test-other. WER 8.40% on Multilingual LibriSpeech (MLS) English [19], WER 8.9% on TEDLIUM2 [20], and WER 6.8% on AISHELL-1 [21] test sets.

Table 1: The accuracy gap between CTC-based models and Seq2Seq and Transducer models, LibriSpeech, WER(%) (See [1, 2] for a comprehensive comparison between CTC, Seq2Seq, and RNN-T models.)

| Model Type | LM | Test Params | Parameters |
|------------|----|-------------|------------|
| QuartzNet-15x5 [9] | CTC | 3.90 | 11.28 | 19 |
| | RNN | 3.90 | 11.28 | 19 |
| Transformers [11] | CTC | 2.80 | 7.10 | 270 |
| | RNN | 2.10 | 4.70 | 270 |
| LAS+SpecAugm [11] | Seq2Seq | 2.80 | 6.80 | 360 |
| | RNN | 2.50 | 5.80 | 360 |
| Transformers [11] | Seq2Seq | 2.89 | 6.98 | 270 |
| | RNN | 2.33 | 5.17 | 270 |
| Conformer-Transducer [12] | CTC+Seq2Seq | 2.10 | 4.90 | 115 |
| | Transducer | 2.88 | 6.60 | 159 |
| | RNN | 2.00 | 4.60 | 159 |
| ContextNet-L [14] | Transducer | 2.10 | 4.60 | 112 |
| | RNN | 1.95 | 4.10 | 112 |
| Conformer-L [15] | Transducer | 2.10 | 4.30 | 118 |
| | RNN | 1.90 | 3.90 | 118 |

2. Model architecture

Citrinet is a 1D time-channel separable convolutional CTC model with a QuartzNet [9] like architecture enhanced with 1D Squeeze-and-Excitation (SE) context modules. Fig. 1 describes the Citrinet-BxRx: C model, where B is the number of blocks, R is the number of repeated sub-blocks per block, and C is the number of filters in the convolution layers of each block. Citrinet uses the standard acoustic front-end: 80-dimensional log-mel filter banks with a 25ms window and a stride of 10ms.

The network starts with a prolog block B0, then three mega-blocks B1, B2, B3, B4, B5, B6, B7, B8, B9, B10, B11, B12, B13, B14, B15, B16, B17, B18, B19, B20, B21, and an epilog module B22. Each mega-block begins with a 1D time-channel separable convolutional layer with stride 2, so Citrinet progressively down-samples the input three times in the time domain. A mega-block is combined from residual blocks B1. Each residual block consists of basic QuartzNet blocks, repeated R times, plus an SE module in the end. A QuartzNet block is composed of 1D time-channel separable convolution with kernel K, batch-norm, ReLU, and dropout layers. All convolutional layers have the same number of channels C except the epilog. Citrinet supports a range of kernel layouts for 1D convolutional layers shown in Table 2. Section 4.1 explains how these kernel layouts were derived. Narrow K1 layout is better for streaming, while

Table 2: Citrinet: kernel layout configurations. Numbers below are kernel size for 1D convolutional layers.

| K | H0 | H1 | H2 | H3 | H4 | H5 | H6 | H7 | H8 | H9 | H10 | H11 | H12 | H13 | H14 | H15 | H16 | H17 | H18 | H19 | H20 | H21 | H22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| K1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K4 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K6 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K7 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| K8 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |

Preprint. Submitted to INTERSPEECH-21

[1] See [1, 2] for a comprehensive comparison between CTC, Seq2Seq, and RNN-T models.

NVIDIA, USA

{smajumdar,jbalam,ohrinchuk,vlavrukhin,vnoroozi,bginsburg}@nvidia.com

arXiv:2104.01721v1  [eess.AS]  5 Apr 2021
wider layouts are designed for offline mode. We use \( K_4 \) as a baseline throughout the paper.

Citrinet uses the Squeeze-and-Excitation (SE) mechanism from [13] to extend model context. SE block computes a learned global scaling factor per channel \( \theta(x) \)

\[
\theta(x) = \sigma(W_2(ReLU(W_1 \tilde{x} + b_1) + b_2)),
\]

where \( W_1, W_2, b_1 \) and \( b_2 \) are trainable weights, \( \sigma \) is the sigmoid, and \( \tilde{x} = \frac{1}{T} \sum_t x_t \). The scaling factor \( \theta(x) \) is applied via pointwise multiplication to each of the input channels:

\[
SE(x) = \theta(x) \odot x
\]

3. Experiments

3.1. LibriSpeech

We trained five configurations of Citrinet with \( K_4 \) kernel layout from Table 2 and \( R = 5 \) which differed only in the number of channels \( C \), on LibriSpeech (LS) dataset [18]. We used a word-piece tokenizer with 256 sub-word units built on the LS train set using the Huggingface library [22]. The same tokenizer is used also for two external language models (6-gram and Transformer) for prefix beam re-scoring. Both LMs are constructed using the default LM corpus combined with text from the LS train transcriptions. To train Citrinet models we used the NovoGrad optimizer [23] with learning rate (LR) of 0.05, \( \beta_1 = 0.8, \beta_2 = 0.25 \) and weight decay of 0.001. We used a cosine annealing LR policy with 1K step linear warmup. We used SpecAugment [11] with two frequency masks (\( F = 27 \)), two time masks for small models (\( C = 256, 384 \)) and ten time masks for other models with 5% adaptive mask length. The models have been trained for 1000 epochs on 32 V100 GPUs with a batch size of 32 per GPU.

The results are shown in Table 3. We use for comparison two Transducers: ContextNet and Conformer. The largest Citrinet model has greedy WER 6.2% test-other set. With LM rescoring, the gap between Citrinet and SOTA Transducers for test-other is reduced to only 0.59 – 0.79%.

3.2. MLS

We trained Citrinet-1024 on the English part of MLS dataset [19]. The original MLS-English train set has 44.66k hours. We filtered out utterances with misaligned transcripts, typos from optical character recognition and ended up with 42.97k hours. To make the model compatible with the LibriSpeech alphabet, we replaced all hyphens with a space character in all splits (train, dev, test). Additionally, we dropped six utterances with Spanish characters from the dev set and one utterance with Russian characters from the test set. Finally, we deleted three instances of the period character from the test set. With LM rescoring, the gap between Citrinet and SOTA Transducers for test-other is reduced to only 0.59 – 0.79%.

| Model          | LM    | Test clean | Test other | Params, M |
|----------------|-------|------------|------------|-----------|
| ContextNet-L   | RNN   | 2.10       | 4.00       | 112.7     |
| Conformer-L    | RNN   | 2.10       | 4.30       | 118       |
| Citrinet-256   | 6-gram| 3.65       | 8.06       | 9.8       |
| Citrinet-384   | 6-gram| 2.94       | 6.71       | 21.0      |
| Citrinet-512   | 6-gram| 2.40       | 6.08       | 36.5      |
| Citrinet-768   | 6-gram| 2.15       | 5.11       | 81        |
| Citrinet-1024  | 6-gram| 2.52       | 6.22       | 142       |

Table 3: LibriSpeech: Citrinet vs Transducers, WER(%)
3.3. TED-LIUM 2

We trained two Citrinet-1024 models for the TED-LIUM 2 corpus [20] with 207 hours of speech. The first model was trained from scratch (TS), and the second model was fine-tuned (FT) from the model pre-trained on MLS. We used 1024 sentence piece tokens trained on the training set of the TED-LIUM 2 for the TS model and used the tokenizer trained on MLS data for the FT model. For the LM (N-grams and Transformer), we used both TED-LIUM 2 text for LM and the texts from the train part. The TS model was trained for 1000 epochs using the NovoGrad optimizer with a peak LR of 0.05. Fine-tuning was done for 200 epochs using a peak LR of 0.005. Both models were trained on 16 GPUs with a batch size of 32 per GPU. All other hyper-parameters are the same as the LibriSpeech model. For comparison, we used two models: RWTH hybrid HMM-based model trained with SpecAugment [25], and end-to-end model composed from 7-layer time-delay NN combined with 3 LSTM layers [26].

Table 5 shows the Citrinet-1024 evaluation on TED-LIUM 2 dev and test sets. The Citrinet-1024 fine-tuned from the pre-trained on MLS model matches the hybrid HMM model [25] and sets a new SOTA for end-to-end NN-based models.

### Table 5: TED-LIUM rev.2: Citrinet-1024 trained from scratch (TS) and fine-tuned (FT) from MLS English, WER (%)

| Model              | LM     | dev | test |
|--------------------|--------|-----|------|
| Dense TDNN-LSTM    | N-gram | 7.60| 8.10 |
|                    | RNN    | 7.10| 7.70 |
| RWTH Hybrid HMM    | 4-gram | 6.80| 7.30 |
|                    | Transf | 5.90| 6.60 |
| Citrinet-1024/TS   | -      | 7.99| 8.90 |
| Citrinet-1024/FT   | 6-gram | 7.75| 7.47 |
|                    | 6-gram | 6.69| 6.05 |

3.4. AISHELL-1

We trained two Citrinet-1024 models for AISHELL-1 [21], a Mandarin corpus with 150 hours of training speech. The first model was trained from scratch (TS). For the second model (FT) weights of the encoder were pre-initialized from the encoder of the English model trained on MLS. Unlike the English language models, we used character level tokenization instead of sentence piece tokenization for Mandarin ASR. The AISHELL-1 TS and FT models were trained with the same parameters as the TED-LIUM 2 models from Section 3.3.

Table 6 compares the Character Error Rate (CER) for Citrinet and two hybrid CTC-attention models: ESPNet Transformer [27] and U2 model [28] composed from Conformer encoder and two decoders - CTC and Transformer. Citrinet-1024 fine-tuned from the English model, has better CER than Transformer-based Seq2Seq model [27].

### Table 6: AISHELL-1: Citrinet-1024 trained from scratch (TS) and Citrinet-1024 fine-tuned (FT) from the model pretrained on MLS English, CER (%)

| Model              | LM     | dev | test |
|--------------------|--------|-----|------|
| ESPNet Transformer | 6.00   | 6.70|      |
| U2 Conformer-Transformer | - | - | 4.72 |
| Citrinet-1024/TS   | -      | 6.16| 6.82 |
| Citrinet-1024/FT   | 4-gram | 5.89| 6.39 |

4. Ablation study

We perform an ablation study to determine the contribution of each component to the model accuracy. We use Citrinet-BxRxC with $B = 21$, $R = 5$, $C = 384$ and kernel layout $K_4$ as baseline. For brevity, we denote the configuration of Citrinet-BxRxC as Citrinet-C, such that it expands to Citrinet-21x5xC. As a baseline we use 1024 Word Piece Tokens for the tokenizer vocabulary. The ablation study is done on the LibriSpeech dataset. All models have been trained for 400 epochs on 32 GPUs with a batch of 32 per GPU. We used the NovoGrad optimizer with betas of (0.8, 0.25), cosine learning rate decay with peak LR of 0.05, and weight decay of 0.001.

#### 4.1. Kernel width

We use as a baseline the wide $K_4$ kernel layout in Table 7.

### Table 7: Citrinet: baseline kernel layout.

| Block | Filters, K |
|-------|------------|
| $B_0$ | 5          |
| $B_1 - B_6$ | 11,13,15,17,19,21 |
| $B_7 - B_{13}$ | 13,15,17,19,21,23,25 |
| $B_{14} - B_{21}$ | 25,27,29,31,33,35,37,39 |
| $B_{22}$ | 41         |

We scale all kernels except the prologue and epilogue layers with the same scaling factor, $\gamma$: the scaled kernel widths are $k'_i = [k_i + \gamma_i]$, where $k_i$ is the original kernel width for the $i$-th block. If $k'_i$ width is even, it is incremented by 1. Table 8 shows the model WER for $\gamma = 0.25, 0.5, 0.75, 1$.

### Table 8: Accuracy vs kernel size ($\gamma$ - kernels scaling factor). Citrinet-384, LibriSpeech, greedy WER (%).

| Kernel scaling factor | dev | clean | other | clean | other |
|-----------------------|-----|-------|-------|-------|-------|
| $K_1$                 | 0.25| 3.36  | 8.89  | 3.56  | 9.28  |
| $K_2$                 | 0.50| 3.26  | 8.59  | 3.41  | 8.74  |
| $K_3$                 | 0.75| 3.15  | 8.75  | 3.43  | 9.04  |
| $K_4$                 | 1.00| 3.49  | 9.31  | 3.62  | 9.28  |

We found that there is an optimal kernel width, and too narrow or too wide kernels lead to higher WER. The narrow kernel layout corresponding to $\gamma = 0.25$ is a primary candidate for
streaming ASR. One can get slightly better accuracy by doubling its receptive field. Increasing the kernel size further will decrease the model accuracy.

4.2. Scaling network in width and depth

We start the scalability study by changing the number of channels $C$ while keeping $R = 5$ and the kernel layout as in Table 7. Table 9 compares the WER and number of parameters for Citrinet with $C = 256$, 384 and 512 channels:

| $C$   | dev clean | other clean | dev other | other M  |
|-------|-----------|-------------|-----------|---------|
| 256   | 4.38      | 11.41       | 4.57      | 11.54   | 10.2 |
| 384   | 3.49      | 9.31        | 3.62      | 9.28    | 21.1 |
| 512   | 3.16      | 8.71        | 3.27      | 8.83    | 37.2 |

Models can be also scaled in depth by changing the number of sub-blocks $R$ per block. The capacity of the model, its depth and the receptive field of the network grow with the number of repeated blocks, increasing the model accuracy. Table 10 compares WER for Citrinet-384 with various values of $R$.

| $R$   | dev clean | other clean | dev other | other M  |
|-------|-----------|-------------|-----------|---------|
| 2     | 4.22      | 11.16       | 4.39      | 11.14   | 11.6 |
| 3     | 3.59      | 10.07       | 3.81      | 9.94    | 14.9 |
| 4     | 3.49      | 9.52        | 3.68      | 9.48    | 18.1 |
| 5     | 3.49      | 9.31        | 3.62      | 9.28    | 21.1 |

4.3. Tokenizer lexicon size

The simple way to reduce the memory usage and increase training and inference speed is to compress intermediate activations along the time dimension. But down-sampling in time is limited by CTC loss which requires the output of the acoustic model to be longer than the target transcription. For example, char-based English CTC models work best at 50 steps per second [1, 29]. To bypass this constraint, we use word-piece encoding [22] or byte-pair encoding to represent the input text using fewer encoded tokens. Using sub-word encoding we are able to train models with 8x contraction on the time dimension. Table 11 compares the effect of the lexicon size on the model accuracy.

| Lexicon Size | dev clean | other clean | dev other | other M  |
|--------------|-----------|-------------|-----------|---------|
| 128          | 3.50      | 9.64        | 3.63      | 9.42    |
| 256          | 3.30      | 8.88        | 3.46      | 8.92    |
| 512          | 3.35      | 9.53        | 3.77      | 9.52    |
| 1024         | 3.49      | 9.31        | 3.62      | 9.28    |
| 2048         | 3.81      | 9.99        | 4.11      | 10.59   |
| 4096         | 7.09      | 14.73       | 7.18      | 14.44   |

The very large lexicon sizes tend to significantly harm transcription accuracy. On the other side, we observed that if the lexicon size is less than 128, CTC loss cannot be computed on a significant number of transcripts that exceed the length of the output of the acoustic model after 8x compression in the time domain.

4.4. Context window size

To study the effect of the context window size of the SE module, we used Citrinet-512 with 1024 Word Piece tokenization. We replace the global pooling operator with an average pooling operator where the pooling size defines the local context windows of 256, 512, and 1024. The results are shown in the Table 12. The context provided by the SE mechanism significantly contributes to the Citrinet accuracy, similar to [14].

| SE window | dev clean | other clean | dev other | other M  |
|-----------|-----------|-------------|-----------|---------|
| -         | 3.39      | 9.05        | 3.36      | 9.09    |
| 256       | 3.01      | 8.59        | 3.36      | 8.61    |
| 512       | 2.96      | 8.12        | 3.27      | 8.26    |
| 1024      | 2.91      | 8.03        | 3.15      | 8.04    |
| Global    | 2.86      | 8.02        | 3.12      | 7.99    |

5. Conclusions

In this paper, we introduced Citrinet - a new end-to-end non-autoregressive CTC-based model. Citrinet enhances the QuartzNet [9] architecture with the Squeeze-and-Excitation mechanism from ContextNet [14]. Citrinet significantly reduces the gap between non-autoregressive and state-of-the-art autoregressive Seq2Seq and RNN-T models [1, 2]. Contrary to the common belief that “CTC requires an external language model to output meaningful results” [30], Citrinet models demonstrate very high accuracy without any external language model on LibriSpeech, MLS, TED-LIUM 2, and AI-SHELL datasets.

The models and training recipes have been released in the NeMo toolkit [31].

6. Acknowledgments

The authors thank the NVIDIA AI Applications team for the helpful feedback and review.

7. References

[1] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, “Exploring neural transducers for end-to-end speech recognition,” in ASRU, 2017.
[2] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in Interspeech, 2017.
[3] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in ICML, 2006.

https://github.com/NVIDIA/NeMo
