Improving Text-Independent Speaker Identification by Introducing Multiplicative layer to Convolutional Layout

Sergey Skrebnev

Independent Researcher
sergey.skrebnev@gmail.com

August 18, 2022

Abstract

In this paper, solving task of text-independent speaker identification, modification of traditional neural network architecture based on regular convolutional networks proposed by adding multiplicative layer similar to one in self-attention. Proposed architecture allows to improve results for sequential data both for smaller datasets like TIMIT (630 speakers) and larger datasets like Voxceleb2 (5994 speakers).

Keywords: *Speaker recognition, speaker identification*

I. Introduction

Speaker identification is a process of identifying person already known to system by listening to either some predefined phrase (text-dependent identification, like calling 'Hey, Siri') or arbitrary speech (text-independent). Implementing this process allows to simplify machine-human interaction and adds more recognition capabilities for security systems. This process is possible due to differences in human vocal tract that make us sound different, each voice with it's own acoustic features [1].

The research on speaker recognition (identification, verification) is dating back to 1960s. Since then, number of approaches to acoustic features were developed: linear predictive cepstral coefficients (LPCC), perceptual linear prediction coefficient (PLP) [2], mel-frequency cepstral coefficients (MFCC) [3]. Later Gaussian Mixture Model with Universal Background Model (GMM-UBM) [4] / i-vectors [5] were proposed, becoming to-go solution until recent ascent of neural networks. In such systems, neural network produces single-dimensional vector as an output called embedding (also known as d-vector). Similar to i-vectors, d-vectors represent utterances in a fixed dimensional space. One of the most intuitive approaches for retrieving acoustic features from utterance into vector form is to treat spectrogram of the utterance as an image and apply some visual-based neural network, like CNN (convolutional neural network). Recently ViTs (Visual Transformers) [6][7] proved themselves most accurate in computer vision tasks, but they are computationally demanding due to high number of parameters. This paper demonstrates approach to create mix of two architectures: CNNs and Tranformers, based on intuition that accuracy, achieved by transformers, comes from ability to model functions via basic multiplication operation between sets of vectors, suchwise it may be possible that multiplication, when added to regular convolutional block, can improve modeling capabilities of the convolutional networks.

II. Rationale

Convolutional networks were best available solution for visual classification tasks before visual transformers (ViTs) were introduced. ViTs reached and currently holding new state of art results, due to ability to model advanced functions through attention mechanism. But what attention is? If we are talking about multiplicative attention in Transformers, it is often described

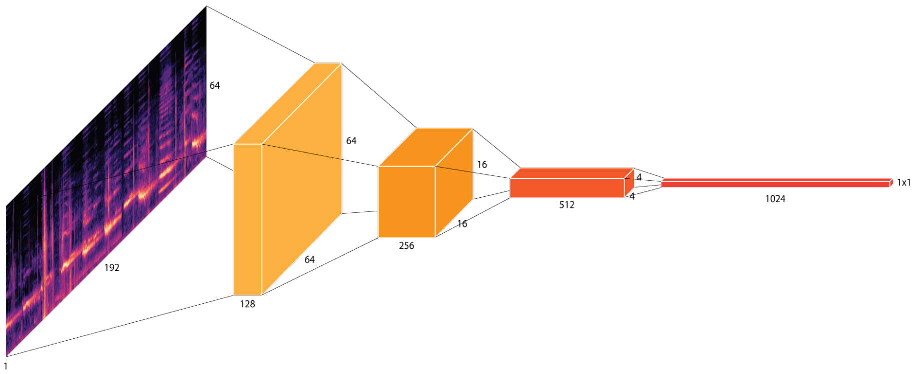

Figure 1: *Convolutional wireframe*

as a number of parallelized multiplications between sets of vectors: multiply **Q**ueries by **K**eys, then apply results to **V**alues.

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Transformers, according to current state-of-art achievements, demonstrate that these blocks are able to model functions at more precise level than sequences of linear and convolutional layers, activation functions added. In some models it is not even necessary to have knowledge transfer between Queries/Keys and Values, for example BERT [8] model is a mix of (only self-)attention blocks (Q==K==V) and feed-forward layers, yet it proved successful. Self-attention is the particular case of attention when there is no knowledge transfer, due to (Q==K==V). This fact may lead to assumption that modeling capabilities are hidden in very basic part of attention: matrix multiplication. Common DNN layers like Dense(Linear) and Convolutional produce linear outputs; combined with recently most popular activation function, ReLU, which, in it's turn, creates non-linearity by combining two linear functions: 0 if $x < 0$ and x if $x \geq 0$, final composition will behave as an intricate set of linear functions. On the contrary, multiplication of inputs by themselves is a non-linear operation producing power of two outputs, with chained multiplications increasing power further. Multiplicative non-linearity, combined with this intricate, but still linear at ε-neighborhood (where ε depends on number of parameters/layers) model of traditional ReLU-based CNN, neural model can produce better results. Intuitively, combined model is similar to Taylor series in its approximation approach.

Based on this intuition, we can try to improve accuracy by introducing multiplicative part to existing convolutional networks architecture. Traditionally, since Alexnet [9], convnets are constructed using two basic principles: gradually increasing number of feature planes (channels) while decreasing dimensions of the image. There are ways to improve accuracy through residual connections, bottleneck and inverted bottleneck blocks, but in the end convnets are transforming image data into a single-dimensional vector. Usually convolutional block consists of:

- convolutional layer
- normalization layer
- activation function

In between these blocks some extra layers can be used, for example pooling. According to assumption, in addition to these layers weighted multiplication with optional (via trainable weight) identity was introduced (see **Fig. 2**). Identity bypass was added with trainable weight to allow model choose better pathway in case it's more beneficial to use identity instead of weighted multiplication.

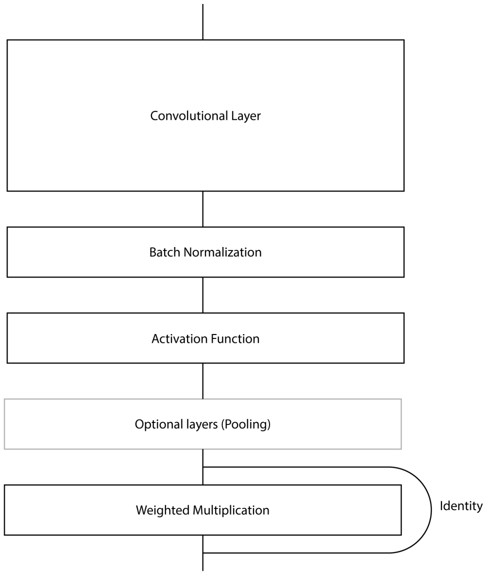

Figure 2: *Extended convolutional block*

$$X = \begin{pmatrix} x_{11} & x_{12} & .. & x_{1n} \\ x_{21} & x_{22} & .. & x_{2n} \\ .. & .. & .. & .. \\ x_{n1} & x_{n2} & .. & x_{nn} \end{pmatrix}, \omega = \begin{pmatrix} \omega_{11} & \omega_{12} & .. & \omega_{1n} \\ \omega_{21} & \omega_{22} & .. & \omega_{2n} \\ .. & .. & .. & .. \\ \omega_{n1} & \omega_{n2} & .. & \omega_{nn} \end{pmatrix}$$

$$WM = X \times X^T \cdot \omega \tag{2}$$

Weighted multiplication, [×]

Self-attention block in original paper, 'Attention is all you need', was designed to work with directional data, sequences of vector-encoded words in text. Same directionality can be observed in spectrograms, where each column is a set of measurements at different frequency bands and these columns are computed apart at specific time intervals. (see **Fig. 3**).

This directionality affects accuracy: with 192 measurements, 64 filter banks per each (array size 64x192), correct order of multiplication is $X \times X^T$. Transposing first element instead drops accuracy even below basic level (experiments showed Top1 accuracy dropping to approximately 67%)

$$x = (1 - \omega_{res}) * x + \omega_{res} * [\times] \tag{3}$$

Semaphore-like trainable identity bypass

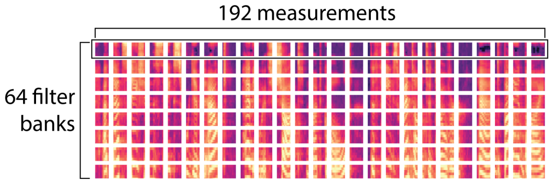

Figure 3: *Spectrogram layout*

III. Method

Datasets

Two datasets were used to run experiments with: TIMIT [10] (630 speakers in dev+test sets) and Voxceleb2 (5994 speakers). TIMIT is acoustic-phonetic continuous speech corpus in 8 dialects of American English, balanced between male and female voices, recorded with little or no noise, and Voxceleb2 [11], corpus created by extracting utterances from video uploaded to Youtube. Voxceleb2 is a good source of real-life utterances, sometimes incorrectly labeled (set of utterances belong to single speaker may contain dialogues or even different speakers per different audio samples), thus accuracy can never achieve same levels as with clean datasets like TIMIT. *Since identification task requires different utterances and not different speakers, TIMIT corpus was reformatted by combining test and dev set into single pool and extracting random audio samples for the newly created test set.*

Preprocessing

Initially each audio sample is preprocessed: converted to 16kHz mono stream; then stripped of intervals considered silent, including those in between words. To define edges between silent and non-silent intervals, threshold of 30db below peak power is chosen. Each audio sample is split into smaller slices by using sliding window 1.92 seconds long with 10ms shift (reason behind these values is purely mathematical, since experiments showed little difference between slices with length varying from 1.8 up to 3 seconds). Records shorter than 0.96s ($\frac{1}{2}$ out of 1.92s) are discarded, remaining are right-padded with zeros. Out of this pool, some number of random slices per each sample are selected, and each slice is converted into melspectrogram with 64 filter banks between 20 and 8000Hz. With frame equal to 25ms and hop of 10ms melspectrograms generated have dimensions of 192 by 64 data points each (ratio 3:1 to simplify further calculations) (see **Fig.3**). Sampling strategy for training is choosing up to 50 random samples per speaker with up to 10 random spectrograms per each sample; strategy for validation is up to 20 random samples per speaker with up to 5 random spectrograms per each. Standard SGD optimizer used with lr=1e-3 and momentum=0.9.

Feature extractor

Identification process in neural network architectures can be divided into two main parts:

1. Extracting features from preprocessed utterances
2. Using either classifier or metric learning approach to identify person

Feature extractor is a simple funnel-like convolutional layout (**Fig 1**), with conv layer with 7x7 kernel at top level, similar to resnets, average pooled with kernel (1,3) to transform 64x192 input to square shape. Following blocks are common convolutional blocks with 3x3 kernel, batch norm and activation function (either ReLU or GeLU). Each block doubles number of channels and cuts input dimensions in twice. As a result, n-dimensional vector created per each slice.

Identification

Selecting between metric learning and regular classifier, second option was chosen due to being less computationally demanding. Of course it is always possible to replace classifier with m-dimensional vector via transfer learning.

Input shape	Block description
1x64x192	Conv2d 7x7/1, AvgPool2d (1,3), [×]
128x64x64	Conv2d 3x3/2, AvgPool2d (2,2), [×]
256x16x16	Conv2d 3x3/2, AvgPool2d (2,2), [×]
512x4x4	Conv2d 3x3/2, AvgPool2d (2,2)
1024 > 630	Linear

Table 1: *Combined model layout, or Just Another NETwork, short for Janet ([×] denotes multiplication layer)*

Model	Parameters	ops*
default, TIMIT	6.85M	30.34G
+ multiplication, TIMIT	6.85M	30.87G
default, VoxCeleb2	12.35M	30.69G
+ multiplication, VoxCeleb2	12.35M	31.22G

*Total mult-adds

Table 2: *Model parameters comparison*

Evaluation

Optimal criterion for evaluation from practical standpoint would be to use full utterance, even if it spans over multiple slices. In this case longer utterances can achieve better score due to noise introduced to VoxCeleb2 dataset by random appearance of additional voices captured within dialogues, overlapped voices or even samples mistakenly added with completely different speakers (this was encountered few times), instead strictest criterion applied: accuracy calculated against each slice, randomly chosen from audio samples' pool during process of generating spectrograms.

Augmentations

With modern networks and their practical applications, augmentations became indispensable in achieving state-of-art results. For the purposes of this article, some most resulting augmentations were implemented: random erase and cutmix/mixup. In addition to data augmentations, label augmentations were also added via label smoothing.

1. Cutmix

   Cutmix/Mixup [12] is a strategy of combining two images from different classes, either

CutMix MixUp

CutMix & MixUp

simply partially covering one image with another (cutmix), or by overlaying one image above another based on opacity (mixup).

2. Random Erase

Random Erase

Random Erase [13] is a strategy of erasing random part of image during training, usually rectangular, filling empty space with constant values.

3. Label Smoothing

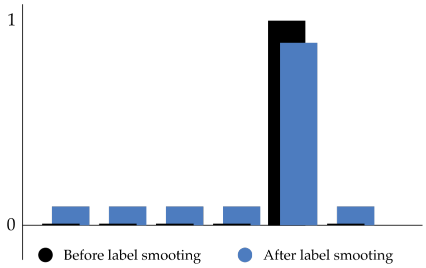

Figure 4: *Label Smoothing*

Label smoothing improves classification accuracy by introducing noise to class labels.

IV. Results

Although model with original convolutional layout demonstrated 99.12% Top1 accuracy for TIMIT dataset, modified layout with multiplication layer was able to improve it further, to 99.65% **without signs of overfitting** to initial data. Another advantage of combined layout is stable convergence attributed to convnets without any needs to run warmup steps using custom learning rate.

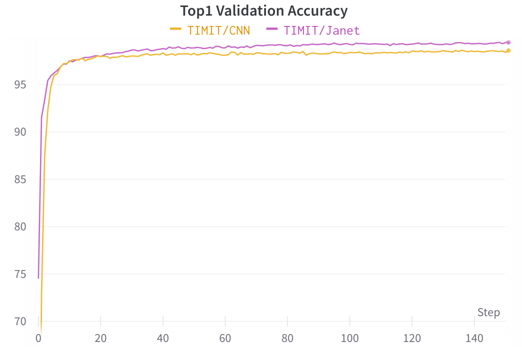

Figure 5: *Top1 validation accuracy, TIMIT*

Layout	Top1 error	Speed*
default convolutional	0.88%	68.4
+ multiplication	0.35%	48.1

*Speed in batches per second, using single RTX 3090

Table 3: *Top1 validation error for TIMIT dataset (630 speakers), best of 300 epochs, batch size = 32*

As for VoxCeleb2, modified model achieved 92.95% Top1 accuracy, improving basic result of 91.18%.

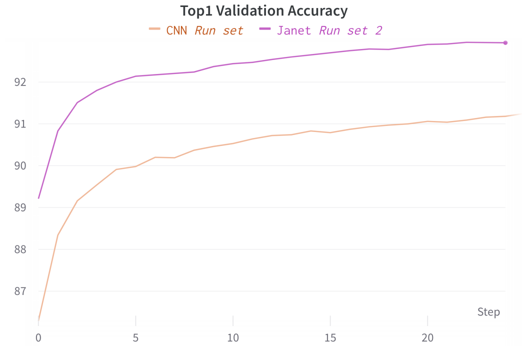

Figure 6: *Top1 validation accuracy, VoxCeleb2*

Layout	Top1 error	Speed*
default convolutional	8.82%	61.4
+multiplication	6.85%	45.2

*Speed in batches per second, using single RTX 3090

Table 4: *Top1 validation error for Voxceleb2 dataset (5994 speakers), 25 epochs, batch size = 32*

V. Conclusion

Models based on multiplicative attention proved to excel convnets. These models were originally designed based on intuition that attention mechanics can detect relations between elements spaced apart at significant distance between each other in sequences. Underlying multiplication can be the reason why these models are so exceptionally good, since this operation produces intricate connections between inputs compared to regular perceptron. This intuition was applied to sequential data in audio samples: to detect relations between measurements across the time span, by adding simple weighted multiplication. Results show that this merge is possible and possesses some benefits: stable conversion, no signs of overfitting and, of course, improved modeling capabilities for both rather simple (TIMIT with 630 speakers) and large (VoxCeleb2 with 5994 speakers) datasets. Theoretically it is possible to apply same ops to regular image classification models like ResNets, if images' data can be sequentially reorganized similar to 'patches' approach in Visual Transformers.

Source code is available at https://github.com/skrbnv/janet

References

[1] A. Lammert, M. Proctor, A. Katsamanis, S. Narayanan, "Morphological Variation in the Adult Vocal Tract: A Modeling Study of its Potential Acoustic Impact", Interspeech 2011

[2] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," the Journal of the Acoustical Society of America, vol. 87, no. 4, pp. 1738–1752, 1990.

[3] P. Mermelstein, "Distance measures for speech recognition, psy- chological and instrumental," Pattern recognition and artificial in- telligence, vol. 116, pp. 374–388, 1976.

[4] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," Digital signal processing, vol. 10, no. 1-3, pp. 19–41, 2000.

[5] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification", IEEE Trans. ASLP, vol. 19, pp. 788798, May 2010.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ICLR 2021, arXiv:2010.11929 [cs.CV]

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention Is All You Need", arXiv:1706.03762 [cs.CL]

[8] J. Devlin, M. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805 [cs.CL]

[9] A. Krizhevsky, I. Sutskever, and G. E Hinton. "Imagenet classification with deep convolutional neural networks". In Advances in Neural Information Processing Systems, 2012.

[10] J. S Garofolo, L. Lamel, M. Fisher, J. G. Fiscus, "TIMIT Acoustic-phonetic Continuous Speech Corpus", ResearchGate link

[11] J. Son Chung, A. Nagrani, A. Zisserman, "VoxCeleb2: Deep Speaker Recognition", arXiv:1806.05622 [cs.SD]

[12] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, Y. Yoo, “CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features“, arXiv:1905.04899 [cs.CV]

[13] Z. Zhong, L. Zheng, G. Kang, S. Li, Yi Yang, “Random Erasing Data Augmentation“, arXiv:1708.04896 [cs.CV]