IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

VOICE ACTIVITY DETECTION

[1]Shilpa Sharma, [2]Shubhanshu Mathur, [3]V Sekhar

[1]Assistant Professor, [2]Student, [3]Student
[1]Department of Computer Science,
[1]Lovely Professional University, Jalandhar, India

Abstract: A data-driven approach of voice activity detection is presented. Voice activity detection (VAD) is the task of recognizing which parts of an audio contains speech and background noise. It is an important and must step to determine which samples to send to the decoder and when to switch the microphone off. Automatic speech recognition (ASR) systems typically need to associate its degree always-on low-complexity Voice Activity Detection (VAD) module to acknowledge the voice before forwarding it to additional process in order that you'll be able to scale back power consumption. In several real-life eventualities, recorded audio is rackety and deep neural networks incontestable additional strong to noise compared to the historically used applied mathematics ways. The analysis study investigates the general potency of 3 distinct low-complexity architectures – particularly Long Short Term Memory (LSTM) Recurrent Neural Networks (RNN), Gated repeated Unit (GRU) RNNs , and implementation of DenseNet. what is more, the influence of Focal Loss (FL) within the Cross-Entropy (CE) criterion throughout coaching is explored and findings square measure compared with recent VAD analysis.

***Index Terms* – Voice Activity Detection, Automatic Speech Recognition, Recurrent Neural Network,**

Introduction Voice activity detection (VAD), often known as speech activity detection, is a crucial initial step in several speech-based systems. It is necessary for Automatic Speech Recognition (ASR), to refrain from word insertions resulting from noise and background speech; it's also employed in audio coding to avoid wasting bandwidth, as well as in multi-party conference systems, for instance, to decrease the quantity of background noise. Early approaches to VAD were dependent on simple energy thresholds or pitch and zero-crossing rate rules. These approaches work effectively in settings when there is little or negligible background noise. Newer approaches consider more sophisticated parameters like autoregressive (AR) model parameters and line spectral frequencies (LSPs). Among the most promising approaches in highly corrupted conditions which happens to be data-driven methods, in which a classifier is trained to predict speech vs. non-speech from acoustic features . Still, the overall efficiency of such approaches degrades when background noise with spectral characteristics identical to speech is present. Very recent studies suggest the fact that the use of long-span features clearly greatly improves robustness in real-life and noisy settings due to the reason that the decision for any frame can be executed in the instance of the prior frames**.** These days many system like smart home devices and speakers – uses ASR as an integral portion of graphical user interface. As embedded systems usually have limited power and are generally battery-driven, there exists a need of low-complexity ASR systems that might be robust to noise. VAD will be the task of determining whether a voice is at the moment present or do not and is typically applied to be the starting point in real-time, always-on ASR systems as a way to decrease overall power consumption by avoiding further processing if no voice exists. To give a good user experience it is crucial for a VAD module to have a very low FRR, which is challenging in noisy environments (i.e. most real-life scenarios). If speech is momentarily rejected by the VAD module, the following step in the processing pipeline may fail to recognize e.g. a wake-word employed to wake up the full ASR system. On the other hand, if a VAD module achieves an infinitesimal FRR but has a high FAR, the power consumption of the VAD process may very well exceed the amount of power it was supposed to save. Historically, VADs have been built using statistical methods but they tend to perform poorly in noisy environments. Google WebRTC VAD is a publicly available and widely used VAD built as a Gaussian Mixture-Model. Modern VADs are typically based on deep neural networks as they have proven far superior to statistical methods in less ideal environments. One research group has applied a LSTM-RNN to VAD in popular Hollywood movies with great success . Recent efforts shift their attention to deep dilated CNNs . Using a 36-layer dilated CNN, they achieve a FAR of 5.67% compared to 6.66% for a comparable LSTM-RNN with a fixed FRR of 1% for both architectures. Although such deep architectures are shown to produce excellent results for VAD they are impractical for use in embedded real-time ASR systems as they are immensely resource-intensive due to their large size. In this paper we briefly present the two mentioned deep learning approaches and introduce the reader to the production of a large dataset built from open sources, FL – an alteration to the CE criterion – is introduced in the following section and the paper then proceeds to present three distinct deep learning architectures proposed in this study.

I. STATE OF ART

1.1 Eyben's approach

The VAD proposed by Florian Eyben, et. al. [2] is based upon a recurrent network architecture using LSTM cells in two distinct setups: 1)one particular recurrent layer and 2)3 recurrent layers. Instead of applying the model to raw audio, 18 MFCCs are extracted making use of a frame size of 25 ms as well as their derivatives are computed to generate a full of 36 acoustic features. Continuous but varied noise is matched to the dataset comprising 26 hours of informal speech. Computed frames are inputted towards the network in sequences of length 50 (equivalent to a 1.25 s window). The precise large number of parameters considering the two networks. They noticed that their approach outperforms statistical methods undoubtedly in relation to accuracy – especially on samples which contain noise. The tiny and large networks achieve an AUC of .951 and .961, respectively, on their test set in comparison with a mere .821 by the best-performing statistical method proposed.

2.2 Shuo-Yiin's approach

Shuo-Yiin, et. al. [3] explore temporal modeling utilizing a CNN for VAD to quickly attain state-of-the-art results. In contrast to [2], they compute 40 MFCCs (also making use of a frame size of 25 ms) but refrain to utilise their derivatives. Therefore, [3] uses a total of 40 acoustic features. Much like the previous study, varied background noise implemented to clean speech to produce a simulated noisy dataset. In this study, a surprising 18.000 hours of audio is used as training data and evaluation is carried out on 15 hours of similar data. Computed frames are inputted to the network in sequences of length 270 (equal to 6.75 s). By optimizing dilation, gating and residual connections they build an enormous 36-layer deep CNN (and an additional fully-connected layer and an output layer) containing approximately 400.000 parameters. They find that their model achieves a FAR of 5.67% compared to 9.76% for a conventional CNN for a fixed FRR at 1%. Additionally, they build a stacked LSTM of similar size, which achieves a FAR of 6.66% under the same circumstances. As such, they reinforce the idea that both convolution and recurrent networks can be used to capture temporal patterns in acoustic data. An important drawback from their proposed architectures; however, is that they are immensely resource-intensive feasibly be used to capture voice in a real-time scenario without introducing a considerable amount of latency.

II. PROPOSED ARCHITECTURES

Three distinct architectures for VAD have already been adopted within this study: an LSTM-RNN, a GRU-RNN with features obtained from convolutional layers, and compact implementation of DenseNet [11]. To evaluate model performance within a constrained parameter environment, all three architectures have been evaluated at two fixed parameter counts: 10.000 and 30.000 parameters (approximately). Therefore, a total of 6 distinct low-complexity models are proposed and of course the influence of parameter space is investigated. The two network sizes are known as large and small in the following.

2.1 Long Short-Term Memory cells

The LSTM-RNN models function as a performance baseline and comprises one unidirectional LSTM-layer (30 cells) along with a couple of fully connected (FC) layers, based on network size. The LSTM cell architecture adopted in this particular study is as originally proposed by [12] and as implemented by [2]. Inputs are formatted in a way that a cell input xk is a 24-dimensional feature vector being made of the MFCCs and deltas equivalent to the kth frame (time-step).

2.2 Gated Recurrent Units and Gated Convolution

The next architecture is a GRU-RNN consisting of 3-4 gated convolutional layers (counting on network size), just one unidirectional GRU-layer (30 cells) and ultimately just one or two FC layers followed up by a softmax output layer. A GRU implements its gating mechanisms similarly to LSTM but lacks a memory unit, which exposes the full hidden state to the cell. Empirically, GRUs have already been confirmed to perform on par with LSTM on a range of sequence modeling problems within the audio domain and the lack of memory units decreases the number of parameters as well as computational complexity. No reduction in performance was found on the challenge addressed in this study, thus GRUs were favored over LSTM cells.

One-dimensional convolution with zero-padding and fixed kernel size of 3 (90 ms) is applied along the temporal axis to capture short-term patterns in time. Thus, each feature suggests a channel in the convolutional layer. Gated convolution is similar to the gating mechanisms present in LSTMs and GRUs in that an input is convoluted independently with two different sets of filters – regular filters and gate filters. The hyperbolic tangent function is applied to output from regular filters and output from gates pass through a sigmoid function. The Hadamard product of the two resulting matrices is then computed and forwarded as the final output of that layer. The full computation of hidden state hk given input xk−1 and weights Wf,k−1, Wg,k−1 can thus be written as Equation :

$$hk = \tanh(Wf\ ,\ k-1 * xk-1)\ .(Wg\ ,\ k-1 * xk-1) \qquad (1)$$

Due to the limited number of convolutional layers as well as the fact that shallow CNNs are less prone to vanishing gradients , residual connections were not found to improve model performance and have thus been abandoned. Batch normalization and dropout (p = .2) is applied whenever appropriate.

2.3 DenseNet

The third architecture proposed is the implementation of DenseNet [11] which has been applied to the temporal character of VAD. DenseNet attempts to maximize the essential information flow through the CNN layers by connecting the CNN layers via concatenation and for that reason improve the feed-forward properties considering the CNN layers. The initial layer includes a dilated CNN layer with max-pooling, which makes it possible for a broad capture of the moving window the parameter count low. The output that are caused by the dilated CNN layer is forwarded into two dense blocks connected through a transition layer, which attempts to diminish over-fitting. The output considering the last dense block is traveled through a single CNN layer with max-pooling and at last a single softmax layer. The most important difference between the small- and large DenseNet is the amount of layers in each dense block and the number of channels used in the CNN layers.

III. EXPERIMENT DETAILS

Models on all three noise levels making use of a batch size of 2048, 30 frames (i.e. a window of 900 ms), and 24 features obtained from MFCC as well as their derivatives. Models are trained for 15 epochs using FL as a criterion plus a γ-value of 0 (regular CE) or 2; whichever yields the best accuracy upon the validation set is kept. If the accuracy is the same, $\gamma = 0$ is used. The value of γ is chosen based upon results in [10] as well as a preliminary evaluation. One limitation of our study is that samples of different noise levels are not shuffled prior to training, which may diminish the effect of FL to some extent. Adam with weight decay 1e-5 and initial learning rate of 1e-3 is applied as optimizer for the LSTM-RNN and GRURNN architectures while Stochastic Gradient Descent (SGD) with a learning rate of 1 and momentum 0.7 is applied to DenseNet as it yields better results. Models are evaluated on a test set generated from the same source but with no overlap in speech. For each of the three noise levels, ROC-Curves are computed and the Area Under Curve (AUC) is used as primary performance metric. Additionally, FAR is calculated for fixed FRR at 1% for comparison to [3].

IV. RESULTS

In this section, we present our experimental performance results of all three architectures proposed for VAD in noisy environments. In subsection 4.1, the impact of FL is investigated. Over the next couple of subsections, the focusing parameter γ that yields the best results on a validation set is selected. Table 1 shows the chosen value of γ for each of the proposed architectures and network sizes.

γ-value	LSTM-RNN	GRU-RNN	DenseNet
Small	0	2	2
Large	2	2	2

Table 1: Selected value of focusing parameter γ for each of the proposed architectures and network sizes.

4.1 Results using Focal Loss

Experimentally, it is found that whether FL has a significant impact on performance or not seems to rely on the particular architecture. Table 2 highlights the positive impact of FL on the large LSTM-RNN while no improvement is present on the small network. While the table only shows AUC for the high-noise test set, minor improvements are evident in the other noise levels as well for the large model while the small model seemingly remains unaffected by FL. Further research is needed to discover the exact impact of FL when placed upon VAD, but it shows potential.

LSTM-RNN	CE FL
Small	0.965 0.965
Large	0.962 0.969

Table 2: AUC on highest noise level test set for two sizes of a LSTM-RNN trained with and without FL ($\gamma = 2$).

IJCRT

4.2. Results from parameter constraints

While using AUC as a measure for performance, the two recurrent architectures do not benefit much from increased network size on the test sets without noise keeping little noise. DenseNet, however, consistently improves its AUC across all noise levels clearly as the number of parameters increases. Therefore, it is speculative whether its performance continues to enhance when further increasing network size. On the highest noise level, LSTM-RNN and GRU-RNN also see significant improvements to AUC as their network size is increased, exactly why we can ask the same question. It should be noted that – since labeling is automated and thus imperfect – there is a natural limit to how well the networks can perform on a given test set. As most models achieve an AUC around .992 with no background noise, this might be near the upper limit for performance provided the imperfect labeling.

V. CONCLUSION.

Based on the conducted experiments, it can be determined that CNNs, RNNs and a mixture of the two are viable architectures for low-complexity VAD in noisy environments. MFCCs have been demonstrated to be compact and reliable acoustic features and the proposed architectures are competent accurately classifying voice despite a comparatively small temporal context (0.9 s). The GMM-based WebRTC is a good public API for VAD on clean speech but is of little to no use on noisy audio. Therefore, it may be well suited for automated labeling but not as a low-complexity replacement for deep neural networks in noisy environments. It has been shown experimentally that Focal Loss can have a minor positive impact on performance (AUC of .962 vs. .969 for CE and FL using $\gamma = 2$, respectively, for training of a large LSTM-RNN and testing on a high-noise set) but a larger impact would likely be observed on a dataset with shuffled noise levels and more varied samples in terms of speech and sources of noise. Based on observations made for different network sizes of each of the three proposed architectures it can be concluded that increasing the number of parameters have a positive impact on performance. Increasing network size from 10.000 to 30.000 for a GRU-RNN decreases FAR for fixed FRR at 1% from 4.71% to 3.61% for clean speech, which is a worthwhile improvement. As such, network size and performance is an important trade-off when developing VAD modules. While a combination of CNNs and RNNs yields the best performance results on sets with no noise or low levels of noise, a small LSTM-RNN scores significantly better on high-noise tests with a FAR for fixed FRR at 1% of 48.13% in comparison with 61.13% and 58.14% for GRU-RNN and DenseNet, respectively, of comparable size. Therefore, it can be speculated – given a low-complexity constraint – whether R-CNNs are more robust alternatives to regular CNNs in environments with high levels of noise, while keeping its edge over regular RNNs on lower noise levels.

VI. Acknowledgment

This paper and the research behind it would not have been possible without the exceptional support of our supervisors, Their enthusiasm, knowledge and attention to detail have been an inspiration and kept our work on track and also this research has proved a little costly and we are most thankful to our institution for providing financial support for the project from which this paper grew.

References

[1] Google, “Webrtc,” https://webrtc.org,)

[2] Florian Eyben, Felix Weninger, Stefano Squartini, and Bjrn Schuller, “Real-life voice activity detection withlstm recurrent neural networks and an application to hollywood movies,” 2013.

[3] Anshuman Tripathi, Aron van den Oord, Bo Li, Gabor Simko, Oriol Vinyals, Shuo yiin Chang, and Tara Sainath, “Temporal modeling using dilated convolution and gating for voice-activity-detection,” in ICASSP2018, 2018.

[4] Daniel Povey Vassil Panayotov, Guoguo Chen and Sanjeev Khudanpur, “Librispeech asr corpus,” 2015.

[5] R. Vogt D. Dean, S. Sridharan and M. Mason, “Qutnoise databases and protocols,” 2010.

[6] “Python interface to the webrtc voice activity detector,”2018.

[7] Mohammed Bahoura and Hassan Ezzaidi, “Hardware implementation of MFCC feature extraction for respiratory sounds analysis,” in 2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA). 6 2013, IEEE.

[8] Swapnil D. Daphal and Sonal K. Jagtap, “DSP based improved speech recognition system,” in 2012 Inter- national Conference on Communication, Information &Computing Technology (ICCICT). 10 2012, IEEE.

[9] Haofeng Kou, Weijia Shang, Ian Lane, and Jike Chong, “Optimized MFCC feature extraction on GPU,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. 6 2013, IEEE.

[10] Tsung-Yi Lin, “Focal loss for dense object detection,” 2018.

[11] Gao Huang, “Densely connected convolutional networks,” 12 2016.

[12] Sepp Hochreiter and Jrgen Schmidhuber, “Long shortterm memory,” Neural computation, vol. 9, pp. 1735–80, 12 1997.

[13] Kyunghyun Cho, “Learning phrase representations using rnn encoderdecoderfor statistical machine translation,” 9 2014.

[14] Junyoung Chun, “Empirical evaluation ofgated recurrent neural networkson sequence modeling,” 12 2014.

[15] Diederik P. Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” 12 2014.

[16] Ming Liang and Xiaolin Hu, “Recurrent convolutional neural network for object recognition,” 2015.