FOCAL LOSS AND DOUBLE-EDGE-TRIGGERED DETECTOR FOR ROBUST SMALL-FOOTPRINT KEYWORD SPOTTING

Bin Liu[1,2]*, Shuai Nie*[1]*, Yaping Zhang*[1,2]*, Shan Liang*[1]*, Zhanlei Yang*[1]*, Wenju Liu*[1]

[1] National Laboratory of Patten Recognition, Institute of Automation, Chinese Academy of Sciences, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, China
{bin.liu2015,shuai.nie,yaping.zhang,sliang,zhanlei.yang,lwj}@nlpr.ia.ac.cn

ABSTRACT

Keyword spotting (KWS) system constitutes a critical component of human-computer interfaces, which detects the specific keyword from a continuous stream of audio. The goal of KWS is providing a high detection accuracy at a low false alarm rate while having small memory and computation requirements. The DNN-based KWS system faces a large class imbalance during training because the amount of data available for the keyword is usually much less than the background speech, which overwhelms training and leads to a degenerate model. In this paper, we explore the focal loss for the training of a small-footprint KWS system. It can automatically down-weight the contribution of easy samples during training and focus the model on hard samples, which naturally solves the class imbalance and allows us to efficiently utilize all data available. Furthermore, many keywords of Chinese conversational assistants are repeated words due to the idiomatic usage, such as 'XIAO DU XIAO DU'. We propose a double-edge-triggered detecting method for the repeated keyword, which significantly reduces the false alarm rate relative to the single threshold method. Systematic experiments demonstrate significant further improvements compared to the baseline system.

Index Terms— keyword spotting, focal loss, double-edge-triggered detecting method, speech recognition

1. INTRODUCTION

With the increasing popularity of mobile devices, speech-enabled technologies are becoming more prevalent. Conversational assistants running on smart phones or smart-home sensors try to provide a fully hands-free experience for users. The keyword spotting (KWS) system is a critical component of human-computer interfaces, which detects a specific keyword from a continuous stream of audio and transits between different running states of the device [1, 2, 3]. Due to resource constraints of mobile devices, the proposed KWS system must have a small memory and CPU footprint, while simultaneously providing very high detection accuracy and very low false alarm (FA) rate.

Traditional approaches to KWS are based on Hidden Markov Models (HMMs) and sequence search algorithms [4, 5, 6, 7]. HMMs are utilized to represent both the keyword and background audio. The background model is also called the filler model and can be used to model non-keyword speech, or noise etc. During decoding, Viterbi search is implemented to find the best path in the decoding graph. The system is triggered when the likelihood ratio of the keyword model to the background model exceeds a pre-defined threshold.

An alternative approach to KWS, based on deep neural network (DNN) with no HMM involved, has been shown to significantly outperform the Keyword/Filler HMM system [1, 8, 9, 10]. DNNs are trained to identify sub keyword targets and the posterior handling module calculates a single confidence score according to the frame-level posterior scores. The system fires when the keyword confidence score exceeds a pre-defined threshold. The trade-off between false rejects and false accepts can be implemented by tuning the threshold, which is the key problem to enable satisfactory user experience in practical applications.

However, the amount of data available for the keyword is typically much less than the background speech due to the cost of data acquisition. KWS system faces a large class imbalance during training, which overwhelms the training and leads to a degenerate model. The focal loss [11] is intended to address the class imbalance of dense object detection by adding a dynamically scaled factor to the standard cross-entropy loss. It can automatically down-weight the contribution of easy instances during training and focus the model on hard instances.

In this paper, we explore the focal loss for the training of a small-footprint KWS system, which naturally handles the class imbalance and enables us to efficiently utilize all data obtainable. Experiments show that the focal loss enables us to train a high-accuracy detector that significantly outperforms the alternatives trained with the standard or weighted cross-entropy loss. In addition, there are many repeated keywords in Chinese conversational assistants because of the Chinese idiomatic usage, such as 'XIAO DU XIAO DU' and 'XIAO AI XIAO AI'. The confidence score with the ordering constraint would rise twice if the keyword is repeated. We propose the double-edge-triggered detecting method for the repeated keyword, which significantly reduces the false alarm rate relative to the single threshold method.

The rest of the paper is structured as follows. In Section 2, we review the related work. In Section 3, we describe the proposed DNN-based KWS system. We present the experiments and conclusions in Section 4 and Section 5, respectively.

2. RELATED WORK

Deep learning-based KWS systems have been widely used due to their superior performance [1]. Multi-style training [12], automatic gain control (AGC) [13] and multi-task learning [14] are proposed in

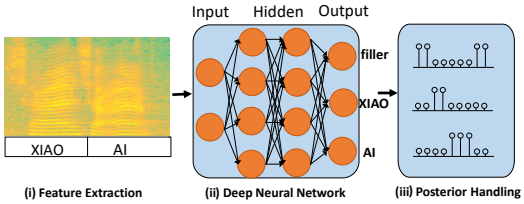


Fig. 1. Framework of Deep KWS system, components from left to right: (i) Feature Extraction (ii) Deep Neural Network (iii) Posterior Handling

order to improve system robustness. KWS systems face a large class imbalance during training. A common solution is to use the class-weighted cross-entropy loss [14], perform some form of hard negative mining [15] [16] or sampling/reweighing schemes [17]. Focal Loss [11] is proposed to deal with class imbalance of dense object detection, which can automatically down-weight the contribution of easy examples during training and focus the model on hard examples. During the posterior handling, [12] defines a keyword score which takes into account the relative order in which the keyword targets are uttered.

3. DNN-BASED KEYWORD SPOTTING SYSTEM

A block diagram of our DNN-based KWS system is shown in Figure 1. Conceptually, our system consists of three modules: (1) a feature extraction module which extracts acoustic features and inputs this into a neural network, (2) a DNN which computes posterior probabilities of each word in the keyword phrase, and (3) a posterior handling module which calculates a single confidence score according to the frame-level posterior probabilities and makes a decision whether the keyword is detected.

3.1. Focal Loss for DNN Training

Suppose that the keyword to be detected, $\mathbf{w}$, consists of M words, $\mathbf{w} = \{w_1, w_2, \cdots, w_M\}$ and w_0 represents a word that does not belong to any of the words in the keyword (denoted as 'filler' in Figure 1). For each frame, t, in the input speech, a feature vector denoted by x_t is fed into the neural network and the posterior probability of the k-th label is denoted by $p^{w_k}(x_t)$. The network parameters are usually trained to optimize a cross-entropy criterion. If the class label is one-hot form, the cross-entropy criterion becomes the negative log-likelihood criterion. For the simplicity of notation, we consider loss functions for a single frame. The cross-entropy loss is given by:

$$\mathcal{L}_{\text{CE}}(\mathbf{\Phi}) = -\log p^{w_{\hat{k}}}(x_t, \mathbf{\Phi}), \tag{1}$$

where $\mathbf{\Phi}$ are the parameters of the DNN, $p^{w_{\hat{k}}}(x_t, \mathbf{\Phi})$ is the output of the final softmax layer corresponding to the class label $w_{\hat{k}}$.

In the case of KWS, the amount of data available for the keyword is normally much less than the background speech and non-speech due to the cost of data acquisition. If we use all the available data for training and don't filter out background data, the positive samples that belong to one of the keywords will be much less than the negative samples that do not belong to any of the words in the keyword. This imbalance leads to an inefficient training as most training data are easy negatives that contribute no useful learning signal. And

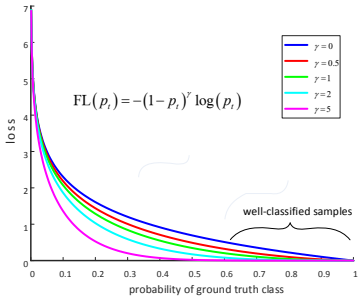


Fig. 2. The visualization of Focal Loss that adds a factor $(1-p_t)^{\gamma}$ to the cross-entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified samples, putting more focus on hard, misclassified samples.

easy negative samples can overwhelm training and lead to a degenerate model. A simple method for addressing the class imbalance is to weight the loss function, i.e., give a higher weight for a frame if the label of the frame belongs to the keyword. More generally, we define a weight vector $\boldsymbol{\alpha}$ with elements $\alpha_k > 0$ defined over the range of class labels k ($w_k \in \{w_0, w_1, \cdots, w_M\}$). We write the α-balanced CE loss as:

$$\mathcal{L}_{\text{WCE}}(\mathbf{\Phi}) = -\alpha_{w_{\hat{k}}} \log p^{w_{\hat{k}}}(x_t, \mathbf{\Phi}). \tag{2}$$

In practice $\boldsymbol{\alpha}$ may be set the inverse of class frequencies or treated as hyper-parameters to be set by cross validation.

The large class imbalance faced during training overwhelms the cross entropy loss. Easily classified negative samples consist of the majority of the loss and dominate the gradient back propagation. While $\boldsymbol{\alpha}$ balances the importance of positive/negative samples, it does not make the difference between easy/hard samples. Instead, we use the focal loss to automatically down-weight easy examples and thus focus training on hard examples.

More formally, we add a dynamically scaling factor to the cross entropy loss, with tunable focusing parameter $\gamma \geq 0$. The focal loss is defined as:

$$\mathcal{L}_{\text{FL}}(\mathbf{\Phi}) = -(1-p_{t,k})^{\gamma}\log(p_{t,k}), \tag{3}$$

where $p_{t,k} = p^{w_{\hat{k}}}(x_t, \mathbf{\Phi})$, is the posterior probability of the corresponding label.

Figure 2 is a visualization of the focal loss for several values of $\gamma \in [0, 5]$. As $p_{t,k} \to 1$, the modulating factor tends to 0 and the loss for well-classified sample is down-weighted. When a sample is misclassified and $p_{t,k}$ is small, the scaling factor is near to 1 and the loss is unaffected. The focusing parameter γ smoothly modifies the rate at which easy samples are down-weighted. When $\gamma = 0$, FL is equivalent to CE. And as γ is increasing, the effect of the modulating factor is also increasing.

Intuitively, the modulating factor reduces the loss contribution from easy samples and extends the range in which a sample receives low loss. For instance, with $\gamma = 3$, the loss of a sample classified with $p_{t,k} = 0.9$ is 1000 times lower than CE loss and with $p_{t,k} \approx 0.9536$ it would have 10000 times lower loss. This in turn increases the importance of correcting misclassified samples (whose loss is scaled down by at most 8 times for $p_{t,k} \leq 0.5$ and $\gamma = 3$).

Algorithm 1 Compute the keyword score.

Input: Sliding windows of the smoothed posterior values $s_t(w_k)$, where $1 \leq t \leq T_s, 1 \leq k \leq M$ (T_s is the sliding windows size and M is the number of keyword).
Output: The ordered keyword score

// computing score for the first keyword
$h(1,1) \leftarrow s_1(w_1)$
for $t = 2$ to T_s **do**
  $h(t,1) \leftarrow \max(h(t-1,1), s_t(w_1))$
end for
// computing score for the remaining keywords
for $k = 2$ to M **do**
  $h(k,k) \leftarrow h(k-1,k-1) * s_k(w_k)$
  for $t = k+1$ to T_s **do**
    $h(t,k) \leftarrow \max(h(t-1,k), h(t-1,k-1) * s_t(w_k))$
  end for
end for
return $h(T_s, M)$

3.2. Detecting Keywords using DNN Posteriors

We run our keyword detection algorithm repeatedly over sliding windows of length T_s of the input signal in order to detect keywords from a continuous stream of audio in real time. We denote $\mathbf{x} = \{x_1, x_2, \cdots, x_{T_s}\}$ as one input window over the utterance, including individual frames $x_t \in \mathbb{R}^n$. In our experiments, these frames are equal to filter-banks features, stacked with left and right context features. We compute the smoothed posterior, $s_t(w_k)$, by averaging the posteriors over the previous L frames,

$$s_t(w_k) = \frac{1}{L} \sum_{j=t-L+1}^{t} p^{w_k}(x_j, \mathbf{\Phi}), \tag{4}$$

where $p^{w_i}(x_j, \mathbf{\Phi})$ is the posterior probability of the k-th label at the j frame. Then the smoothed values are used to define the keyword score, $\hat{h}(x, w)$, as follows:

$$\hat{h}(x,w) = \left[\prod_{k=1}^{M} \max_{1 \leq t \leq T_s} s_t(w_k) \right]^{\frac{1}{M}}, \tag{5}$$

where the window sizes T_s and L are hyper-parameters to be set by cross-validation.

The simplicity of the keyword score in (5) is the major advantage. The score can be calculated in $\Theta(MT)$ time, which has been shown to achieve excellent KWS performance. However, this score computation does not take into account the relative order in which the keyword targets are uttered. Therefore, we define another keyword score, $h(x, w)$, as the largest product of the smoothed posteriors in the input sliding window, subject to the constraint that the detected words are uttered in the same order as in the specified keyword,

$$h(x,w) = \left[\max_{1 \leq t_1 < \cdots < t_M \leq T_s} \prod_{i=1}^{M} s_{t_i}(w_i) \right]^{\frac{1}{M}}. \tag{6}$$

Although the keyword score in (6) contains additional constraints, it can still be calculated in $\Theta(MT)$ time using dynamic programming, which is described in Algorithm 1. The following experiments demonstrate that the ordering constraint imposed in (6) significantly reduces FAs relative to the score in (5). All results in this paper are therefore reported using the keyword score in (6).

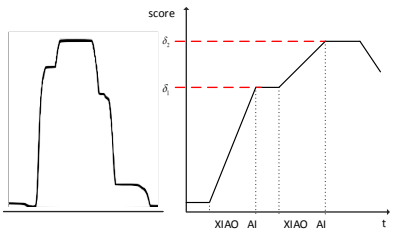


Fig. 3. The keyword score containing relative order constraints if the keyword is repeated. The left side is the real score curve, and the right is the diagram.

3.3. Double-Edge-Triggered Detecting Method

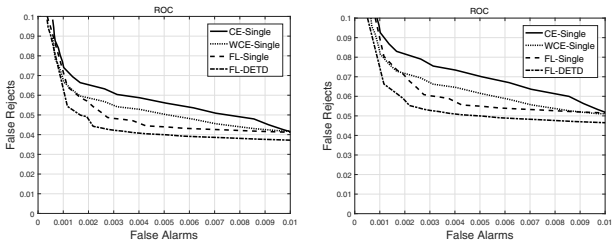
After the keyword confidence score is computed, we usually use a single threshold detecting method. The system fires if the score exceeds a pre-defined threshold. The trade-off between false rejects and false accepts rate can be implemented by tuning the threshold. If the keyword is repeated, such as 'XIAO AI XIAO AI', the keyword score with the ordering constraint will rise twice, which is shown in Figure 3. When a user utters the first part of the keyword, the ordering constraint is partly satisfied and the keyword score rises. When the second part is uttered, the constraint is totally satisfied and the score increases to a higher level. Therefore, we define two thresholds δ_1, δ_2 for the repeated keyword, one is lower that detects the first score rise and the other is higher that detects the second rise. Only if the two rises are simultaneously detected, the system will be triggered. We can adjust two thresholds δ_1, δ_2 to balance the FR and FA rate. The proposed double-edge-triggered detecting method can capture the essential characteristic of the confidence score and significantly reduces the false alarm rate relative to the single threshold method. Suppose the false alarm rate for a single threshold is $\frac{1}{e}$, the FA rate for the double-edge-triggered detecting method is $\frac{1}{e^2}$ ($\frac{1}{e} \gg \frac{1}{e^2}$ as $e \gg 1$).

4. EXPERIMENT

4.1. Datasets

We develop our KWS system for the keyword "XIAO AI XIAO AI" and "XIAO XIN XIAO XIN". In order to evaluate the proposed approach, we collect about 7K utterances containing the keyword "XIAO AI XIAO AI" and 4K utterances containing "XIAO XIN XIAO XIN". We also collect a much larger set of approximately 39K utterances which do not contain any of the keywords and are used as 'negative' training data.

In order to improve system robustness, we perform multi-condition training. Far-field sets are constructed by augmenting the original set with impulse responses corresponding to various configurations. And we artificially corrupt each utterance with a variety of background noises at SNRs randomly sampled between $[0\text{dB}, +10\text{dB}]$, where the noise sounds are sampled from daily-life environments. We also create a multi-speed training set by perturbing the speed of the speech data. The utterances are then randomly



(a) Results on clean speech (b) Results on noisy speech

Fig. 4. ROC curves comparing performance of the system that employs cross-entropy ('CE-Single'), weighted cross-entropy ('WCE-Single'), focal loss ('FL-Single') and double-edge-triggered detecting method ('FL-DETD') for the clean and noisy evaluation sets. Curves closer to the origin are better.

split into training, development, and evaluation sets in the ratio of 80:5:15, respectively. Models are trained in noisy conditions, and evaluated in both clean and noisy conditions.

4.2. Setup

In the following experiments, our DNN models are feed-forward, fully connected neural networks with three hidden layers of 128 neurons, which meets the requirement of a small memory and CPU footprint. We take 40-dimensional filter-banks (computed over 32ms of speech, with a 16ms frame-shift) as the input features, and each dimension of features is normalized to have zero mean and unit variance over the training set. We use 10 frames of left-context and 5 frames of right-context features as the final inputs in order to capture temporal information. We use rectified linear unit (ReLU) activation functions for the hidden layers [18]. The softmax output layer contains output targets for each word in the keyword phrase, plus a single additional output target which represents all frames that do not belong to any of the words in the keyword (denoted as 'filler' in Figure 1). We determine labels for each input frame by performing a forced-alignment using a large LVCSR system [19]. We run the keyword detection algorithm over sliding windows of 100 frames ($T_s = 100$) with posteriors smoothed over 30 frames ($L = 30$).

The baseline system is trained to optimize a cross-entropy criterion (denoted as 'CE') and a weighted cross-entropy criterion (denoted as 'WCE'). Our proposed model is trained by minimizing the focal loss introduced in Section 3.1 (denoted as 'FL'). We use the Adam optimization algorithm [20] for training, with a batch size of 128 and a learning rate of 0.0002. In addition, we compare the proposed double-edge-triggered with the single threshold detecting method during decoding. We calculate confidence scores using Equation (5) and make decisions by the different detecting methods.

KWS performance is measured by plotting a receiver operating curve (ROC), which calculates the false reject (FR) rate per false alarm (FA) rate. Detailed quantitative comparison is given at 0.5% FA rate and our goal is to achieve low FR rates while maintaining low FA rates.

4.3. Results

Firstly, we evaluate the impact of focal loss criterion on system performance. Receiver operating characteristic (ROC) curves comparing the systems of focal loss ('FL-Single'), cross-entropy ('CE-Single') and weighted cross-entropy ('WCE-Single') criterion are presented in Figure 4. We use the single threshold detecting method during decoding. As can be seen in the figure, the focal loss criterion significantly improves performance over the baseline 'WCE-Single' system on the clean and noisy sets, with relative improvements of 15.4% (clean) and 12.5% (noisy) in FR rate at 0.5% FA rate. FL can effectively discount the effect of easy negatives, focusing all attention on the hard negative examples, which naturally solves the class imbalance and allows us to efficiently utilize all data available.

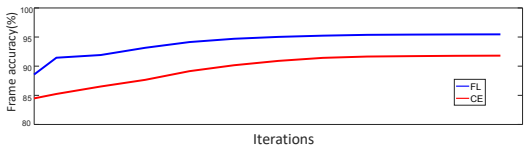


Fig. 5. Frame accuracy during training for focal loss ('FL') and cross-entropy ('CE') criterion.

In order to further make a comparison between focal loss and cross-entropy loss criterion, we analyze the frame accuracy during training. Figure 5 shows that the model trained by focal loss criterion not only arrives at higher frame accuracy but also learns more quickly.

We also investigate the double-edge-triggered detecting method described in Section 3.3. ROC curves comparing the systems of double-edge-triggered ('FL-DETD') and the single threshold ('FL-Single') detecting method are presented in Figure 4. It can be seen that the proposed double-edge-triggered detecting method significantly outperforms the single threshold method. Compared with the single threshold method, the proposed double-edge-triggered detecting method uses two thresholds and detects two rises of the keyword score for the repeated keyword, which significantly reduces the FA rate and improves the system practical performance.

5. CONCLUSIONS

In this paper, we explore the focal loss for the training of a small-footprint KWS system. It can automatically down-weight the contribution of easy samples during training and focus the model on hard samples, which naturally solves the class imbalance and allows us to efficiently utilize all data available. Furthermore, we propose a double-edge-triggered detecting method for the repeated keyword, which significantly reduces the false alarm rate relative to the single threshold method. Systematic experiments demonstrate significant further improvements compared to the baseline system. In the future, we will investigate the voice style transfer using deep learning and develop a KWS system on a smaller dataset in order to reduce the cost of data acquisition.

6. ACKNOWLEDGEMENTS

This work was supported by the China National Nature Science Foundation (No. 61573357, No. 61503382, No. 61403370, No. 61273267, No. 91120303).

7. REFERENCES

[1] Guoguo Chen, Carolina Parada, Heigold, and George, "Small-footprint keyword spotting using deep neural networks," pp. 4087–4091, 2014.

[2] Ming Sun, Varun Nagaraja, Bjrn Hoffmeister, and Shiv Vitaladevuni, "Model shrinking for embedded keyword spotting," in *IEEE International Conference on Machine Learning and Applications*, 2016, pp. 369–374.

[3] Qing He, Gregory W. Wornell, and Wei Ma, "An adaptive multi-band system for low power voice command recognition," in *INTERSPEECH*, 2016, pp. 1888–1892.

[4] R. C Rose, "A hidden markov model based keyword recognition system," *Proc of Icassp Albuquerque Nm Usa*, vol. 1, pp. 129–132 vol.1, 1990.

[5] Jay G Wilpon, Lawrence R Rabiner, Chin Hui Lee, and E. R Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[6] J. G Wilpon, L. G Miller, and P Modi, "Improvements and applications for key word recognition using hidden markov modeling techniques," in *International Conference on Acoustics, Speech, and Signal Processing*, 1991, pp. 309–312 vol.1.

[7] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 627–630 vol.1.

[8] Preetum Nakkiran, Raziel Alvarez, Rohit Prabhavalkar, and Carolina Parada, "Compressing deep neural networks using a rank-constrained topology," 2015.

[9] Tara N Sainath and Carolina Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[10] George Tucker, Minhua Wu, Ming Sun, Sankaran Panchapagesan, Gengshen Fu, and Shiv Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *INTERSPEECH*, 2016, pp. 1878–1882.

[11] Tsung Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollr, "Focal loss for dense object detection," pp. 2999–3007, 2017.

[12] Rohit Prabhavalkar, Raziel Alvarez, Carolina Parada, Preetum Nakkiran, and Tara N. Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4704–4708.

[13] Juan Pablo Alegre Prez, Santiago Celma Pueyo, and Beln Calvo Lpez, *Automatic Gain Control: Techniques and Architectures for RF Receivers*, Springer Publishing Company, Incorporated, 2011.

[14] Sankaran Panchapagesan, Ming Sun, Aparna Khare, Spyros Matsoukas, Arindam Mandal, Bjrn Hoffmeister, and Shiv Vitaladevuni, "Multi-task learning and weighted cross-entropy for dnn-based keyword spotting," in *INTERSPEECH*, 2016, pp. 760–764.

[15] K. K Sung, "Learning and example selection for object and pattern detection," *PhD thesis, MIT AI Lab*, 1995.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg, *SSD: Single Shot MultiBox Detector*, Springer International Publishing, 2016.

[17] Samuel Rota Bulo, Gerhard Neuhold, and Peter Kontschieder, "Loss max-pooling for semantic image segmentation," 2017.

[18] M.D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, and J. Dean, "On rectified linear units for speech processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 3517–3521.

[19] Navdeep Jaitly, Patrick Nguyen, Andrew Senior, and Vincent Vanhoucke, "Application of pretrained deep neural networks to large vocabulary conversational speech recognition," *Proc Interspeech*, 2012.

[20] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.