ADVERSARIAL EXAMPLES FOR IMPROVING END-TO-END ATTENTION-BASED SMALL-FOOTPRINT KEYWORD SPOTTING

Xiong Wang[1*], *Sining Sun*[1*], *Changhao Shan*[1], *Jingyong Hou*[1], *Lei Xie*[1†], *Shen Li*[2], *Xin Lei*[2]

School of Computer Science, Northwestern Polytechnical University, Xi'an, China[1]
Mobvoi AI Lab, Beijing, China[2]

ABSTRACT

In this paper, we explore the use of adversarial examples for improving a neural network based keyword spotting (KWS) system. Specially, in our system, an effective and small-footprint attention-based neural network model is used. Adversarial example is defined as a misclassified example by a model, but it is only slightly skewed from the original correctly-classified one. In the KWS task, it is a natural idea to regard the false alarmed or false rejected queries as some kind of adversarial examples. In our work, given a well-trained attention-based KWS model, we first generate adversarial examples using the fast gradient sign method (FGSM) and find that these examples can dramatically degrade the KWS performance. Using these adversarial examples as augmented data to retrain the KWS model, we finally achieve 45.6% relative and false reject rate (FRR) reduction at 1.0 false alarm rate (FAR) per hour on a collected dataset from a smart speaker.

***Index Terms*—** end-to-end, KWS, adversarial examples, attention

1. INTRODUCTION

Smart devices usually listen to a large amount of audio data generated by users and the surrounding environments. In order to activate the speech interactions between devices and users, a standby keyword spotting (KWS) or wake-up word detection module, is particularly important to detect predefined keyword(s) in audio stream to trigger voice interactions. A good KWS system needs to maintain high robustness with low false rejections and false alarms while being efficient, low power consumption and small-footprint.

Various KWS approaches have been proposed, including large vocabulary continuous speech recognition (LVCSR) based lattice search approaches [1, 2, 3], hidden Markov model (HMM) based keyword/filler approaches [4, 5, 6] and query-by-example (QbyE) based template matching approaches [7, 8]. Recently, with the development of deep learning and its successful applications in speech recognition, deep neural networks (DNNs) have been introduced to KWS [9, 10, 11, 12]. This approach is highly attractive to run on device with small footprint and low latency, as the size of the DNN can be easily controlled and no complicated graph-search is involved. Recently, attention-based end-to-end method has also been introduced to the KWS task [13] and further performance improvement has been observed. Still following the DNN framework, this approach significantly simplifies the mode structure and the decoding procedure.

The performance of a KWS system is typically evaluated by two criteria, FRR and FAR. Although many DNN models achieve superior performance with decent level of low FRR and FAR, the real-world application system can still be falsely triggered when the queries are totally unrelated to the keyword, or be falsely rejected when the queries are keyword obviously. Worsely, the queries triggering false alarm (FA) or false reject (FR) are non-reproducible because of the complicated acoustic environments and many other unpredictable reasons. Thus, this non-reproducible attribute makes it difficult to further improve the KWS performance. It is interesting that such kind of false-alarmed or false-rejected queries can be regarded as *adversarial examples* [14] in the machine learning area.

The concept of adversarial example was first proposed in [15] for computer vision tasks and further developed by many followers, from adversarial example generation [16] to adversarial example defense [17]. Simply speaking, an adversarial example is a misclassified example by a model, but it is only slightly skewed from the original correctly-classified one. These examples can be generated by adding unnoticeable perturbations to the original examples. Recently, audio adversarial examples were also proposed [18], in which the authors tried to generated audio examples that can easily mislead a well-trained speech recognition system. More specifically, given any audio waveform, they can produce another that is over 99.9% similar, but transcribes as any phrase they choose. These studies indicate that the outputs of neural network models are not smooth with respect to inputs and there are "blind spots" in the input space.

In this paper, we explore the use of adversarial examples for improving an attention-based DNN KWS system. This

[*] The first two authors contributed equally to this work.

[†] Lei Xie is the corresponding author.

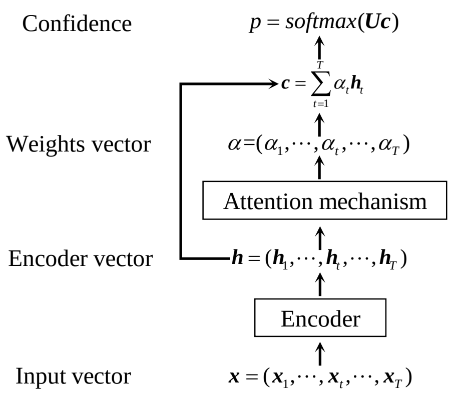

Confidence
$p = softmax(\boldsymbol{Uc})$
$\boldsymbol{c} = \sum_{t=1}^{T} \alpha_t \boldsymbol{h}_t$
Weights vector
$\alpha=(\alpha_1,\cdots,\alpha_t,\cdots,\alpha_T)$
Attention mechanism
Encoder vector
$\boldsymbol{h} = (\boldsymbol{h}_1,\cdots,\boldsymbol{h}_t,\cdots,\boldsymbol{h}_T)$
Encoder
Input vector
$\boldsymbol{x} = (\boldsymbol{x}_1,\cdots,\boldsymbol{x}_t,\cdots,\boldsymbol{x}_T)$

Fig. 1. Attention end-to-end KWS model.

study is motivated by our recent work on robust speech recognition [19], in which, instead of attacking speech recognition systems, we use adversarial examples as a data augmentation method for robust speech recognition during acoustic model (AM) training. As mentioned earlier, introducing adversarial examples into KWS is pretty natural because of the existence of FA and FR queries. Our studies show that there exists such kind of adversarial examples that can apparently trigger FR and FA in a well-trained NN KWS system. Hence following our idea in [20], we further explore the ways to enhance the KWS model using adversarial examples. Finally, we achieve 45.6% relative FRR reduction at 1.0 FAR per hour on a collected dataset from a smart speaker.

The rest of this paper is organized as follows. Section 2 briefly introduces the attention-based end-to-end KWS approach. Section 3 gives details about the generation of adversarial examples. Section 4 shows our experiments and results and Section 5 concludes this paper.

2. ATTENTION MODEL

In this paper, we adopt an recently-proposed attention-based end-to-end KWS model [13], as shown in Figure 1. This simple architecture consists of two modules, an encoder and an attention model. The encoder is usually a recurrent neural network (RNN) which is used to extract representations from the input features. The attention layer transforms the representations into a fixed length vector. s Formally, suppose the input feature sequence of the model is $\boldsymbol{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T)$, and the output sequence of encoder is $\boldsymbol{h} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_t, \cdots, \boldsymbol{h}_T)$, where T is the length of the sequence. This encoder can be expressed as

$$\boldsymbol{h} = Encoder(\boldsymbol{x}) \tag{1}$$

In our work, gated recurrent units (GRUs [21]) are adopted as the encoder. Then the attention model generates a context vector $\boldsymbol{c}$:

$$\boldsymbol{c} = \sum_{t=1}^{T} \alpha_t \boldsymbol{h_t} \tag{2}$$

Specifically, we employ soft attention [22], which is described by the following equations:

$$e_t = \boldsymbol{v}^T tanh(\boldsymbol{W}\boldsymbol{h}_t + \boldsymbol{b}) \tag{3}$$

$$\alpha_t = \frac{\exp(e_t)}{\sum\limits_{t=1}^{T} \exp(e_t)} \tag{4}$$

Finally, we compute the confidence by a projection and a softmax function:

$$p = \mathit{softmax}(\boldsymbol{U}\boldsymbol{c}) \tag{5}$$

where $\boldsymbol{U}$ is linear transformation matrix, and p represents the confidence of the keyword which needs to be detected.

3. ADVERSARIAL EXAMPLES

Adversarial examples can be generated by adding some well-designed small perturbations to the original examples. We call this kind of perturbations as *adversarial perturbations*. How to generate adversarial perturbations attracts lots of interests in computer vision and speech processing fields. In this paper, we use a popular method, the fast gradient sign method (FGSM) proposed by [16] in our KWS system. We would like to verify if this method still works when the input is a time series sequence.

Typically, a DNN with parameters $\boldsymbol{\theta}$ can be represented as a function $f(\boldsymbol{x}, \boldsymbol{\theta})$ with input $\boldsymbol{x}$. Given a well-trained network and a pair of correctly-classified example $(\boldsymbol{x}_i, y_i)$, where y_i is the corresponding ground truth label, the corresponding adversarial example $\boldsymbol{x}_i^{adv}$ can be defined as

$$\boldsymbol{x}_i^{adv} = \boldsymbol{x}_i + \boldsymbol{\delta}_i \tag{6}$$

so that

$$y_i \neq f(\boldsymbol{x}_i^{adv}, \boldsymbol{\theta}) \tag{7}$$

where

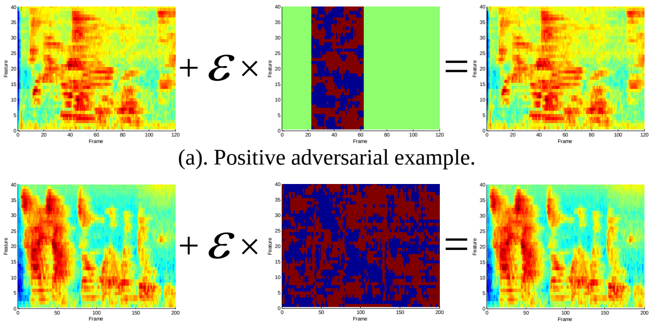
$$\|\boldsymbol{\delta}_i\| \ll \|\boldsymbol{x}_i\| \tag{8}$$

The perturbations satisfying these conditions can interfere the correctness of the original model. Here, FGSM is used to find and generate these perturbations. The idea behind FGSM is pretty straightforward. Given a pair of training example $(\boldsymbol{x}_i, y_i)$ and the loss function $L(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta}))$[1], FGSM tries to find a direction in input space which makes the loss function increase efficiently. This direction can be obtained by deriving $L(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta})$ with respect input $\boldsymbol{x}_i$. So, we have

$$\begin{aligned} \boldsymbol{\delta}_i^{FGSM} &= \varepsilon \operatorname{sign}(\frac{\partial L(y_i, f(\boldsymbol{x}_i, \boldsymbol{\theta}))}{\partial \boldsymbol{x}_i}) \\ \boldsymbol{x}_i^{adv} &= \boldsymbol{x}_i + \boldsymbol{\delta}_i^{FGSM} \end{aligned} \tag{9}$$

where ε is a small constant to adjust the amplitude of the perturbation. Sign function is used here to make it easy to satisfy the constraint in equation 8. At this point, we are ready to produce the adversarial examples for the experiments as described below.

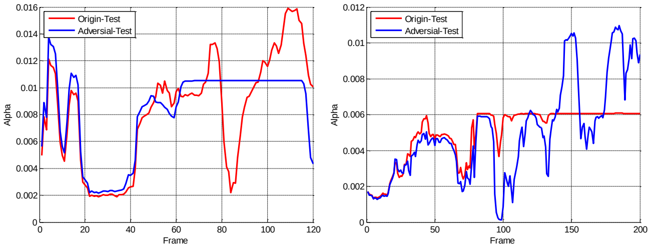
[1] Usually cross entropy loss function is used.



(a). Positive adversarial example.

(b). Negative adversarial example.

Fig. 2. Adversarial queries generation.



(a). Positive adversarial example. (b). Negative adversarial example.

Fig. 3. Visualization of attention layer's weights. The blue line represents the adversarial example's attention weights with $\varepsilon = 0.1$ and the red line represents the original example's attention weights.

4. EXPERIMENTS

4.1. Corpus preparation

We used wake-up data collected from Mobvoi smart speaker TicKasa Fox[2] to verify our KWS approach. The wake-up term is composed of three Mandarin syllables ("hai xiao wen"). Our dataset covers 523 different speakers, including 303 children and 220 adults. In addition, each speaker's collection includes positive utterances (with wake-up word) and negative utterances recorded with different speaker-to-microphone distance and different signal-to-noise (SNR) ratio where noises are from typical home environments. In total, there are 20K positive examples (∼10 hours) and 54K negative examples (∼57 hours) used as the training data. The validation set includes 2.3K positive examples (∼1.1h) and 5.5K negative examples (∼6.2h) while the test set includes 2K positive examples (∼1h) and 5.9K negative examples (∼6h). The speakers involved in each set are not overlapped. 40-dimensional Mel-filterbank is used as acoustic features.

4.2. Experimental setups

In this work, we followed the same model architecture used in [13]. For the encoder, 1-layer RNN with GRUs was adopted. Compared with the positive examples, the negative recordings usually had long duration, so we segmented the negative examples with maximum length of 200 frames (2s) during training. In the test stage, a sliding 200-frame window was applied to the test examples and the window shift size is 1-frame. The KWS system was triggered if at least one segment's score was larger than a pre-set threshold. Our experiments were conducted using TensorFlow and ADAM [23] was the optimizer.

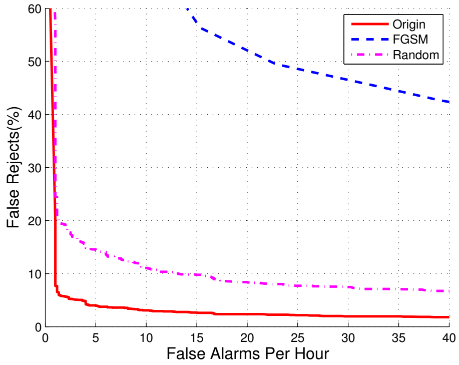


Fig. 4. ROC curves of different perturbed test sets ($\varepsilon = 0.1$). Origin represents the original test set. Random means adding random-sign perturbations to all test queries. FGSM means adding FGSM perturbations to all test queries.

4.3. Adversarial queries generation

Given a well-trained attention-based KWS model, we would like to confirm if adversarial examples can be generated using FGSM. Specifically, we want to generate false-alarmed queries based on negative examples and false-rejected queries based on positive queries using FGSM. If these examples can be generated easily, we can verify that the model is vulnerable to adversarial examples. In other words, the model is not smooth, because a very small perturbation in the input space can lead to a huge change in the output space.

We generated adversarial examples using FGSM on the test set data, as shown in Figure 2. For the positive example perturbation (namely Pos-FGSM), perturbations were only added to the keyword segment, as depicted in Figure 2 (a). As for the negative example perturbation (namely Neg-FGSM), perturbations were added directly to the entire utterance, as depicted in Figure 2 (b). When we tested the attention KWS model using the generated adversarial examples, we found that FAR and FRR increased dramatically, as shown in Figure 4. We analyzed the attention layer's weights of the "bad case" queries before and after adding adversarial perturbations. Figure 3 gives such an example, where the figures depicts attention layer weight changing along the time for a positive example (a) and a negative example (b). We found that even an invisible small perturbation on the spectrum can lead to very obvious changes in the attention layer. It seems that the errors can be accumulated over time because the at-

[2] https://www.chumenwenwen.com/product/tickasa-fox.html

Table 1. Performance of retrained model using different augmentation strategies. FRR is at 1.0 FAR per hour.

Type	Origin	Random	Neg-FGSM	Pos-FGSM	All-FGSM
FRR (%)	7.67	6.59	5.77	**4.17**	5.41
Gain (%)	0	14.1	24.8	**45.6**	29.5

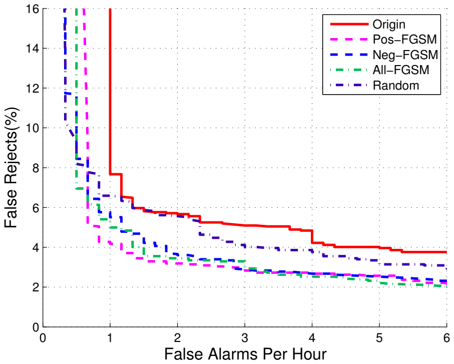


Fig. 5. ROC curves of different data augmentation strategies ($\varepsilon = 0.1$).

tention layer's weights change much faster at the final part of the queries.

As a sanity check, we also tested the random perturbation case. Instead of using the gradient signs to generate the perturbations, random signs ($+1$ or -1) were used. Obviously, from Figure 4, we found that adding random perturbations to the test queries can slightly degrade the model's performance. On the contrast, adding FGSM perturbations to the test queries can severely hurt the model, which means that the model is clearly vulnerable to the adversarial examples. This interesting phenomenon gives us further space to improve the KWS model's performance.

4.4. Training augmentation using adversarial examples

Observations from Section 4.3 show that the current model is very sensitive to adversarial perturbations and the unsmooth problem does exist. In order to improve model robustness, we further augmented training data using adversarial examples. Specifically, we retrained the model using the training strategy proposed by [19]. During the training stage, for every mini-batch data, adversarial examples were generated dynamically (positive and/or negative examples). Then these examples were used to train the model again. In this work, we also tried different augmentation strategies, including augmenting positive queries only, augmenting negative queries only or augmenting all the queries. The model was initialized by a well-trained model using normal training data only.

Figure 5 shows the ROC curves of all the methods with $\varepsilon = 0.1$. Here, Pos-FGSM and Neg-FGSM mean that using positive and negative adversarial examples respectively for data augmentation during training, while All-FGSM and All-Random mean that adding adversarial and random-sign perturbations to all the training data respectively. Table 1 shows the FRR when FAR is at 1.0 on the test set. We can see that the Pos-FGSM and Neg-FGSM based data augmentations can significantly reduce the FRR, with 45.6% and 24.8% relative reduction, respectively. As a comparison, random perturbation-based augmentation slightly improves the performance. In summary, augmenting the training data with adversarial queries is an effective way to improve model robustness.

Table 2. Performance of retrained model with different ε for Pos-FGSM. FRR is at 1.0 FAR per hour.

ε	Origin	0.01	0.10	0.20	0.30
FRR (%)	7.67	5.56	**4.17**	4.69	13.4
Gain (%)	0	27.5	**45.6**	38.9	-74.7

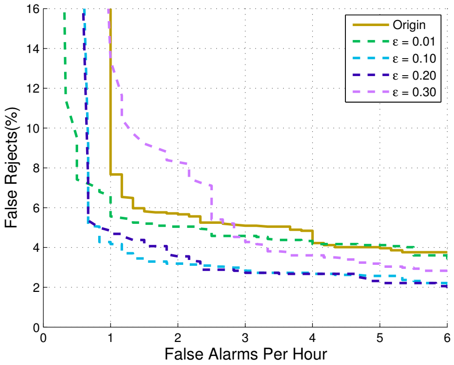


Fig. 6. ROC curves of different ε for Pos-FGSM.

As shown in Figure 6 and Table 2, we also tried different adversarial weights ε for positive adversarial queries augmentation (Pos-FGSM). When $\varepsilon = 0.10$, we can obtain the best result. Larger value, such as $\varepsilon = 0.30$, may degrade the performance because it introduces larger perturbations.

5. CONCLUSIONS

In this paper, we explored the use of adversarial examples for improving the performance of an attention-based end-to-end KWS model. We first verified that false-alarmed and false-rejected queries could be created easily using the FGSM-based adversarial example generation method. Then we augmented the training data using these generated adversarial examples to retrain our attention-based NN KWS model. In summary, we discover that, tested on our corpus, augmenting the training data with adversarial queries is an effective way to improve model robustness. In future, we will test our adversarial data augmentation approach on a larger dataset to examine the performance. Moreover, as speech is sequential data, we plan to take sequential information into consideration and develop new approach to generate adversarial examples specifically for speech data.

6. ACKNOWLEDGEMENTS

The research work is supported by the National Natural Science Foundation of China (No.61571363).

7. REFERENCES

[1] Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan, "Vocabulary independent spoken term detection," in *ACM SIGIR*, 2007, pp. 615–622.

[2] Petr Motlicek, Fabio Valente, and Igor Szoke, "Improving acoustic based keyword spotting using LVCSR lattices," in *ICASSP*, 2012, pp. 4413–4416.

[3] I Fan Chen, Chongjia Ni, Boon Pang Lim, Nancy F. Chen, and Chin Hui Lee, "A novel keyword+LVCSR-filler based grammar network representation for spoken keyword search," in *ISCSLP*, 2014, pp. 192–196.

[4] Binfeng Yan, Rui Guo, Xiaoyan Zhu, and Bo Zhang, "An approach of keyword spotting based on HMM," in *WCICA*, 2000, pp. 2757–2759.

[5] J Robin Rohlicek, William Russell, Salim Roukos, and Herbert Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *ICASSP*, 1989, pp. 627–630.

[6] Ch Choisy, "Dynamic handwritten keyword spotting based on the NSHP-HMM," in *ICDAR*, 2007, pp. 242–246.

[7] Jingyong Hou, Lei Xie, and Zhonghua Fu, "Investigating neural network based query-by-example keyword spotting approach for personalized wake-up word detection in Mandarin Chinese," in *ISCSLP*, 2016, pp. 1–5.

[8] Guoguo Chen, Carolina Parada, and Tara N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *ICASSP*, 2015, pp. 5236–5240.

[9] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *ICASSP*, 2014, pp. 4087–4091.

[10] Zhehuai Chen, Yanmin Qian, and Kai Yu, "Sequence discriminative training for deep learning based acoustic keyword spotting," *Speech Communication*, 2018.

[11] George Retsinas, Giorgos Sfikas, Nikolaos Stamatopoulos, Georgios Louloudis, and Basilis Gatos, "Exploring critical aspects of CNN-based keyword spotting. a PHOCNet study," in *IAPR*, 2018, pp. 13–18.

[12] Santiago Ndez, Alex Graves, J Schmidhuber, and rgen, "An application of recurrent neural networks to discriminative keyword spotting," in *ICANN*, 2009, pp. 220–229.

[13] Changhao Shan, Junbo Zhang, Yujun Wang, and Lei Xie, "Attention-based end-to-end models for small-footprint keyword spotting," in *INTERSPEECH*, 2018, pp. 2037–2041.

[14] Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie, "Domain adversarial training for accented speech recognition," in *ICASSP*, 2018, pp. 4854–4858.

[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[18] Nicholas Carlini and David Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *SPW*, 2018.

[19] Sining Sun, Ching-Feng Yeh, Mari Ostendorf, Mei-Yuh Hwang, and Lei Xie, "Training augmentation with adversarial examples for robust speech recognition," in *INTERSPEECH*, 2018, pp. 2404–2408.

[20] Yusuke Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition.," in *INTERSPEECH*, 2016, pp. 2369–2372.

[21] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Computer Science*, 2014.

[22] Shixiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-End attention based text-dependent speaker verification," in *ISTL*, 2017, pp. 171–178.

[23] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.