NEURAL NETWORKS FOR SUPERVISED PITCH TRACKING IN NOISE

Kun Han and *DeLiang Wang*

Department of Computer Science and Engineering
& Center for Cognitive and Brain Sciences
The Ohio State University
Columbus, OH 43210-1277, USA
{hank,dwang}@cse.ohio-state.edu

ABSTRACT

Determination of pitch in noise is challenging because of corrupted harmonic structure. In this paper, we extract pitch using supervised learning, where probabilistic pitch states are directly learned from noisy speech. We investigate two alternative neural networks modeling the pitch states given observations. The first one is the feedforward deep neural network (DNN), which is trained on static frame-level features. The second one is the recurrent deep neural network (RNN) capable of learning the temporal dynamics trained on sequential frame-level features. Both DNNs and RNNs produce accurate probabilistic outputs of pitch states, which are then connected into pitch contours by Viterbi decoding. Our systematic evaluation shows that the proposed pitch tracking approaches are robust to different noise conditions and significantly outperform current state-of-the-art pitch tracking techniques.

Index Terms— Pitch estimation, Deep neural networks, Recurrent neural networks, Viterbi decoding, Supervised learning

1. INTRODUCTION

Pitch, or fundamental frequency ($F0$), is one of the most important characteristics of speech signals. A pitch tracking algorithm robust to background interference is critical to many applications, including speech separation, and speech and speaker identification [7, 23]. Although pitch tracking has been studied for decades, it is still challenging to extract pitch from speech in the presence of strong noise, where the harmonic structure of speech is severely corrupted.

Previous studies typically utilize signal processing to attenuate noise [4, 6] or statistical methods to model harmonic structure [22, 3, 12], and then determine several pitch candidates for each time frame. The pitch candidates can be connected into pitch contours by dynamic programming [6, 3] or hidden Markov models (HMMs) [22, 13]. However, the selection of pitch candidates is often ad hoc and a hard decision of candidate selection may be less optimal. Instead of rule-based selection of the pitch candidates, we propose to supervisedly learn the posterior probability that a frequency bin is pitched given the observation in each frame. With the probability of each frequency bin, a Viterbi decoding algorithm is utilized to form continuous pitch contours.

A deep neural network (DNN) is a feed-forward neural network with more than one hidden layer [9], which has been successfully used in signal processing applications [16, 21]. In speech recognition, the posterior probability of each phoneme state is modeled by the DNN, which motivates us to adopt the idea for pitch tracking, i.e., we use the DNN to model the posterior probability of each pitch state given the observation in each frame. Further, a recurrent neural network (RNN) is suited for modeling nonlinear dynamics. Recent studies have shown promising results using RNNs to model sequential data [20, 15]. Given that speech is inherently a sequential signal and temporal dynamics is crucial to pitch tracking, it is natural to consider RNNs as a model to compute the probabilities of pitch states.

In this study, we investigate both DNN and RNN based supervised approaches for pitch tracking. With proper training, both DNN and RNN are expected to produce reasonably accurate probabilistic outputs in low SNRs.

This paper is organized as follows. The next section relates our work to previous studies. Section 3 discusses the details of the proposed pitch tracking algorithm. The experimental results are presented in Section 4. We conclude the paper in Section 5.

2. RELATION TO PRIOR WORK

Recent studies on robust pitch tracking explored either the harmonic structure in the frequency domain, the periodicity in the time domain or the periodicity of individual frequency subbands in the time-frequency domain.

In frequency domain, the harmonic structure contains rich

This research was supported in part by an AFOSR grant (FA9550-12-1-0130), an NIDCD grant (R01 DC012048), and the Ohio Supercomputer Center.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

information regarding pitch. Previous studies extracted pitch from spectra of speech, by assuming that each peak in the spectrum corresponding to a potential pitch harmonic [17, 8]. SAFE [3] utilized prominent SNR peaks in speech spectra to model the distribution of the pitch using a probabilistic framework. PEFAC [6] combined nonlinear amplitude compression to attenuate narrowband noise and chose pitch candidates from the filtered spectrum.

Another type of approaches utilizes the periodicity of the speech in the time domain. RAPT [18] calculated the normalized autocorrelation function (ACF) and chose the peaks as the pitch candidates. YIN [4] algorithm used the squared difference function based on ACF to identify the pitch candidates.

A variant of the temporal approach extracts pitch using the periodicity of individual frequency subbands in the time-frequency domain. Wu *et al*. [22] modeled pitch period statistics on top of a channel selection mechanism and used an HMM for extracting continuous pitch contours. Jin and Wang [13] used cross-correlation to select reliable channels and derived pitch scores from a constituted summary correlogram. Lee and Ellis [14] utilized Wu *et al*.'s algorithm to extract the ACF features and trained a multi-layer perceptron classifier on the principal components of the ACF features for pitch detection. Huang and Lee [12] computed a temporally accumulated peak spectrum to estimate pitch.

3. ALGORITHM DESCRIPTION

3.1. Feature extraction

The features used in this study are extracted from the spectral domain based on [6]. We compute the log-frequency power spectrogram and then normalize with a long-term speech spectrum to attenuate noises. A filter is then used to increase the harmonicity.

Specifically, let $X_t(f)$ denotes the power spectral density (PSD) of the frame t in the frequency bin f. The PSD in the log-frequency domain can be represented as $X_t(q)$, where $q = \log f$. Then, the normalized PSD can be computed as:

$$X'_t(q) = X_t(q)\frac{L(q)}{\overline{X}_t(q)} \tag{1}$$

where $\overline{X}_t(q)$ denotes the smoothed averaged spectrum of speech and $L(q)$ represents the long-term average speech spectrum. If there is a strong narrowband noise at frequency q, it will lead to $\overline{X}_t(q) \gg L(q)$ and result in $X'_t(q) < X_t(q)$. In addition, the speech spectral components at other frequencies q' will be enhanced because $X'_t(q') > X_t(q')$. Therefore, the normalized PSD can compensates for speech level changes, but also attenuates narrowband noises.

In the log-frequency domain, the spacing of the harmonics is independent of the period frequency f_0 so their energy can be combined by convolving $X_t(q)$ with a filter with impulse response

$$h(q) = \sum_{k=1}^{K} \delta(1 - \log k) \tag{2}$$

where $\delta(\cdot)$ denotes the Dirac delta function, k indexes the harmonics, and $K = 10$. Due the the width of each harmonic peak will be broadened by the analysis window and the variation of f_0, we use a filter with broadened peaks having an impulse response defined by:

$$h(q) = \begin{cases} \beta - \dfrac{1}{\gamma - \cos(2\pi e^q)}, & \text{if } \log(0.5) < q < \log(K + 0.5) \\ 0, \text{otherwise} \end{cases} \tag{3}$$

where β is chosen so that $\int h(q)dq = 0$, and γ controls the peak width which is set to 1.8. The resulting normalized PSD $X'_t(q)$ is convolved with an analysis filter $h(q)$.

The convolution result $\tilde{X}_t(q) = X'_t(q) \star h(q)$ contains peaks corresponding to the period frequency and its multiples and submultiples. So we have a spectral feature vector in time frame t:

$$\mathbf{Y}_t = (\tilde{X}_t(q_1), \ldots, \tilde{X}_t(q_n))^T$$

Since neighboring frames contains useful information for pitch tracking, we incorporate the neighboring frames into the feature vector. Therefore, the final frame-level feature vector is

$$\mathbf{Z}_t = (\mathbf{Y}_{t-d}, \ldots, \mathbf{Y}_{t+d})^T$$

where d is set to 2 in our study.

3.2. DNN for pitch state estimation

Predicting the posterior probability for each pitch state is important to this study. The first approach we propose is to use a DNN to compute them. To simplify the computation, we quantize the plausible pitch frequency range 60 to 404 Hz using 24 bins per octave in a logarithmic scale, a total of 67 bins [14], corresponding to 67 states $s^1, \ldots, s^{67}$. We also incorporate a nonpitched state s^0 corresponding to an unvoiced or speech-free state. To train the DNN, each training sample is the feature vector $\mathbf{Z}_t$ in the time frame t, and the target is a 68-dimensional vector of pitch states $\mathbf{s}_t$, whose element s_t^i is 1 if the groundtruth pitch is within the corresponding frequency bin, otherwise 0.

The input layer of the DNN corresponds to the input feature vector. The DNN includes three hidden layers with 1600 sigmoid units in each layer, and a softmax output layer whose size is set to the number of pitch states, i.e., 68 output units. The number of hidden layers and the hidden units are chosen from cross-validation. In order to learn the probabilistic output, we use cross-entropy as the objective function. The trained DNN produces the posterior probability of each pitch state i: $P(s_t^i|\mathbf{Z}_t)$.

3.3. RNN for pitch state estimation

The second approach for pitch state estimation is the RNN. An RNN is able to capture the long-term dependencies through connections between hidden layers, which suggests that it can model the pitch dynamics in nature. An RNN has hidden units with delayed connections to themselves, and the activation $\mathbf{h}_j$ of the jth hidden layer in the time frame t is:

$$\begin{aligned} \mathbf{h}_j(t) &= \phi(\mathbf{x}_j(t)) \\ \mathbf{x}_j(t) &= \mathbf{W}_{ji}^T \mathbf{h}_i(t) + \mathbf{W}_{jj}^T \mathbf{h}_j(t-1) \end{aligned} \tag{4}$$

where ϕ is the nonlinear activation function, which is the sigmoid function in this study. $\mathbf{W}_{ji}$ denotes the weight matrix from the ith layer to the jth layer, and $\mathbf{W}_{jj}$ self-connections in the jth layer. Since the recursion over time on $\mathbf{h}_j$, a RNN can be unfolded through time and can be seen as a very deep network with T layers, where T is the number of time steps.

The structure of the RNN in our study is shown in Fig. 1, which includes two hidden layers. Each hidden layer has 256 hidden units and only the units in the hidden layer 2 have self-connections. The input and the output layers are the same as in the DNN.

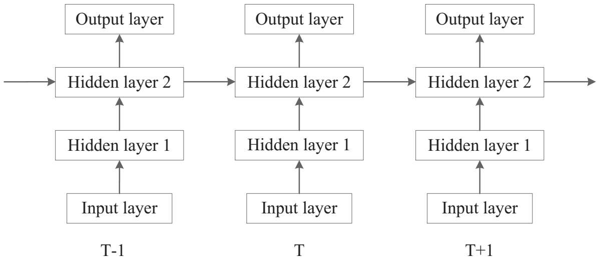
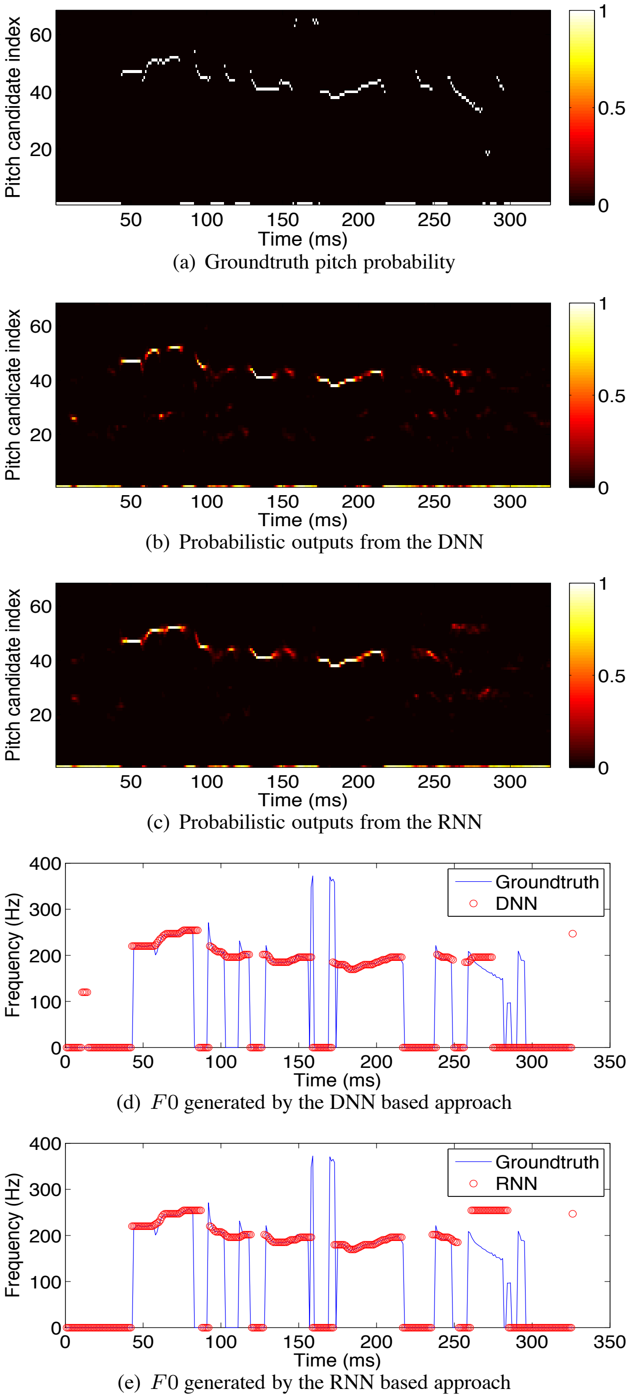


Fig. 1: Structure of the RNN unfolded through time. The RNN has two hidden layers and the hidden layer 2 has the connections to itself.

We use truncated backpropagation through time to train the RNN and the length of each truncation is set to 15 frames. Due to the RNN is trained on sequential features, the output of the RNN in the tth frame is the posterior probability $P(s_t^i|\mathbf{Z}_1, \ldots, \mathbf{Z}_t)$, where the observation is a sequence from the past to the current frame instead of the feature in the current frame.

3.4. Viterbi decoding

The DNN or RNN produces the posterior probability for each pitch state s_t^i. We then use Viterbi decoding [5] to connect those pitch states based on the probabilities. The likelihood used in Viterbi algorithm is proportional to posterior probability divided by the prior $P(s^i)$. The prior $P(s^i)$ and the transition matrix can be directly computed from the training data. Note that, since we train the pitched and nonpitched frames together, the prior of the nonpitched state $P(s^0)$ is usually much larger than that of each pitched state, result-



(a) Groundtruth pitch probability

(b) Probabilistic outputs from the DNN

(c) Probabilistic outputs from the RNN

(d) $F0$ generated by the DNN based approach

(e) $F0$ generated by the RNN based approach

Fig. 2: (a) Groundtruth pitch states. In each time frame, the probability of a pitch state is 1 if it corresponds to the groundtruth pitch; otherwise 0. (b) Probabilistic outputs from the DNN. (c) Probabilistic outputs from the RNN. (d) Pitch contours. The circles denote the pitch generated by the DNN based approach, and solid lines the groundtruth pitch. (e) Pitch contours. The circles denote the pitch generated by the RNN based approach, and solid lines the groundtruth pitch.

ing in that the likelihood of the nonpitched state is relatively small, and Viterbi algorithm may have bias towards pitched states. We introduce a parameter $\alpha \in (0, 1]$ multiplying the prior of the nonpitched state $P(s^0)$ to balance the ratio between the pitched and nonpitched states, which can be chosen from a development set.

The Viterbi algorithm outputs a sequence of pitch states for a sentence. We convert the sequence of pitch states to frequencies and then smooth the continuous pitch contours using moving average to generate the final pitch contours.

Fig. 2 shows pitch tracking results using our approaches. This example is a female utterance mixed with factory noise in -5 dB SNR. Fig. 2 (a) shows the groundtruth pitch states extracted from clean speech using Praat [1]. The probabilistic outputs of the DNN and the RNN are shown in Figs. 2(b) and (c), respectively. Compared with Fig. 2(a), the probabilities of groundtruth pitch states in both Figs. 2(b) and (c) dominate in most time frames. In some time frames (e.g., 100 ms to 120 ms), the RNN yields better probabilistic outputs than the DNN, probably because of its capacity to capture temporal context. Figs. 2 (d) and (e) show the pitch contours after using Viterbi decoding.

4. EXPERIMENTAL RESULTS

To evaluate the performance of our approach, we use the TIMIT database [24] to construct the training and the test set. The training set contains 250 utterances including 50 male speakers and 50 female speakers. The noises used in the training phase include babble noise from [10], factory noise, and high frequency radio noise from NOISEX-92 [19]. Each utterance is mixed with each noise type in three SNR levels: -5, 0, and 5 dB, therefore the training set includes $250 \times 3 \times 3 = 2250$ sentences. The test set contains 20 utterances including 10 male speakers and 10 female speakers. All utterances and speakers are not seen in the training set. The noise types used in the test set include the three training noise types and three new noise types: cocktail-party noise, crowd playgroud noise, and crowd music [11]. We point out that although the three training noise types are included in the test set, the noise recordings are cut from different segments. Each test utterance is mixed with each noise in four SNR levels -10, -5, 0, and 5 dB.

The groundtruth pitch is extracted from the clean speech using Praat [2]. We evaluate the pitch tracking results in terms of two measurements: detection rate (DR) on the voiced frames, i.e., a pitch estimate is considered as correct if the deviation of the estimated $F0$ is within $\pm 5\%$ of the groudtruth $F0$. Another measurement is the voicing decision error (VDE) [14] indicating how many percentage frames are misclassified in terms of pitched and nonpitched:

$$\text{DR} = \frac{N_{0.05}}{N_p}, \quad \text{VDE} = \frac{N_{p \to n} + N_{n \to p}}{N} \tag{5}$$

Here, $N_{0.05}$ denotes the number of frames with the pitch frequency deviation smaller than 5% of the groundtruth frequency. $N_{p \to n}$ and $N_{n \to p}$ denote the number of frames misclassified as nonpitched and pitched, respectively. N_p and N are the number of pitched frames and total frames in a sentence.

We compare our approaches with three state-of-the-art pitch tracking algorithms: PEFAC [6], Jin and Wang, [13], and Huang and Lee [12]. As shown in Fig. 3, both the DNN and the RNN based approaches have substantially higher detection rates than other approaches. The advantages hold for both seen noise and unseen noise conditions, demonstrating that the proposed approaches generalize well to new noises. Note that, both DNN and RNN also significantly outperform other approaches in -10 dB SNR condition, which is not included in the training set. The RNN performs slightly better than the DNN, and the average advantages to other approach are greater than 10%.

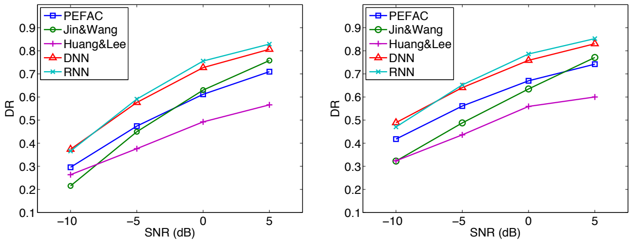


Fig. 3: (a) DR results for seen noises. (b) DR results for new noises.

Fig. 4 shows the VDE results. Since Huang and Lee's algorithm does not produce pitched/nonpitched decision, we only compare our approaches with PEFAC and Jin and Wang. The figure clearly shows that our approaches achieve better voicing detection results than others.

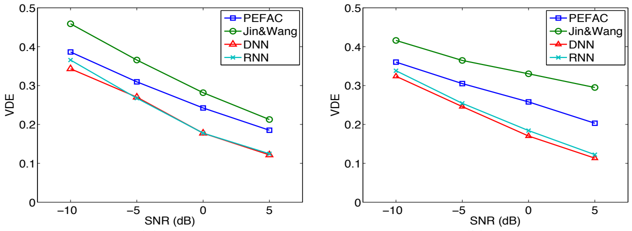


Fig. 4: (a) VDE results for seen noises. (b) VDE results for new noises.

5. CONCLUSION

We have proposed to use neural networks to estimate the posterior probabilities of pitch states for pitch tracking in noisy speech. Both DNNs and RNNs produce very promising pitch tracking results. In addition, they also generalize well to new noisy conditions.

6. REFERENCES

[1] P. Boersma and D. Weenink. (2007) PRAAT: Doing Phonetics by Computer (version 4.5). [Online]. Available: http://www.fon.hum.uva.nl/praat

[2] ——, *PRAAT: Doing Phonetics by Computer (version 4.5)*, 2007, http://www. fon.hum.uva.nl/praat.

[3] W. Chu and A. Alwan, "SAFE: a statistical approach to F0 estimation under clean and noisy conditions," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 3, pp. 933–944, 2012.

[4] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, p. 1917, 2002.

[5] G. D. Forney Jr, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[6] S. Gonzalez and M. Brookes, "A pitch estimation filter robust to high levels of noise (PEFAC)," in *Proc. EUSIPCO 2011*, 2011.

[7] K. Han and D. L. Wang, "A classification based approach to speech segregation," *J. Acoust. Soc. Am.*, vol. 132, no. 5, pp. 3475–3483, 2012.

[8] D. J. Hermes, "Measurement of pitch by subharmonic summation," *J. Acoust. Soc. Am.*, vol. 83, p. 257, 1988.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[10] G. Hu, "100 nonspeech sounds, 2006," http://www.cse.ohio-state.edu/pnl/corpus/HuCorpus.html.

[11] ——, "Monaural speech organization and segregation," Ph.D. dissertation, The Ohio State University, Columbus, OH, 2006.

[12] F. Huang and T. Lee, "Pitch estimation in noisy speech using accumulated peak spectrum and sparse estimation technique," *IEEE Trans. Speech, Audio Process.*, vol. 21, no. 3, pp. 99–109, 2013.

[13] Z. Jin and D. L. Wang, "HMM-based multipitch tracking for noisy and reverberant speech," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1091–1102, 2011.

[14] B. S. Lee and D. P. W. Ellis, "Noise robust pitch tracking by subband autocorrelation classification," in *Proc. of Interspeech*, 2012.

[15] A. L. Maas, Q. V. Le, T. M. O'Neil, O. Vinyals, P. Nguyen, and A. Ng, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. of Interspeech 2012*, 2012.

[16] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, 2012.

[17] M. R. Schroeder, "Period histogram and product spectrum: New methods for fundamental-frequency measurement," *J. Acoust. Soc. Am.*, vol. 43, p. 829, 1968.

[18] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[19] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, no. 3, pp. 247–251, 1993.

[20] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Proc. of ICASSP 2012*. IEEE, 2012, pp. 4085–4088.

[21] Y. Wang and D. L. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, 2013.

[22] M. Wu, D. L. Wang, and G. J. Brown, "A multipitch tracking algorithm for noisy speech," *IEEE Trans. Speech, Audio Process.*, vol. 11, no. 3, pp. 229–241, 2003.

[23] X. Zhao, Y. Shao, and D. L. Wang, "CASA-based robust speaker identification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 5, pp. 1608–1616, 2012.

[24] V. Zue, S. Seneff, and J. Glass, "Speech database development at MIT: TIMIT and beyond," *Speech Communication*, vol. 9, no. 4, pp. 351–356, 1990.