Automatic Recognition of Pitch Movements Using Time-Delay Recursive Neural Network

Sung-Suk Kim[1], Mark Hasegawa-Johnson[2], and Ken Chen[2]

1 Introduction

Words that a talker considers semantically or pragmatically important are often produced with a fundamental frequency contour called a pitch accent. A pitch accent is an unusually high F0 (possibly a local maximum) or an unusually low F0 (possibly a local minimum) designed to draw attention to the important word [1]. The presence of a pitch accent correlates with other changes in the acoustic signal, including increased duration of vowels [2] and increased burst amplitude and voice onset time (VOT) of stop consonants [3]. Knowledge of pitch accent placement would therefore be useful prior information for automatic speech recognition.

This paper proposes a novel method for the automatic recognition of pitch accents with no prior knowledge about the phonetic content of the signal (no knowledge of word or phoneme boundaries or of phoneme labels). In the framework presented here, the problem of pitch accent recognition is considered to be a special case of the general problem of context-dependent, non-parametric dynamic contour recognition. The recognition problem is non-parametric because the distribution of F0 is unknown; in particular, there is no evidence that the distribution of F0 is Gaussian. The recognition problem is context-dependent because F0 encodes much more than just prosody: in particular, talker dependence, dependence on speaking style, and short-time acoustic phonetic information encoded in the F0 trajectory must be ignored. The recognition algorithm used in this paper is a time-delay recursive neural network (TDRNN) [8]. A TDRNN is a neural network classifier with two different representations of dynamic context: delayed input nodes allow the representation of an

[1]School of Computer and Information, Yong-In University, South Korea; sskim@yongin.ac.kr
[2]ECE Department, University of Illinois at Urbana-Champaign; {jhasegaw,kenchen}@uiuc.edu

explicit trajectory $F0(t)$, while recursive nodes provide long-term context information that can be used to normalize the input trajectory. Section 2 of this paper describes a selection of papers in the field of prosody-dependent speech recognition, and briefly discusses the importance of the problem. Section 3 describes the TDRNN architecture used in these experiments. Section 4 describes the experimental methods used for the recognition of pitch accents. Section 5 gives the results, and Section 6 presents conclusions.

2 Background

Prosodic labels are potentially useful in automatic speech understanding systems for at least four reasons. First, prosody correlates with syntax: Price et al. [14] showed that prosody may be used to disambiguate syntactically distinct sentences with identical phoneme strings, while Kim et al. [7] have demonstrated that prosody may be used to infer punctuation of a recognized text. Second, prosody correlates with meaning. Third, prosody is useful for the detection and subsequent processing of speech disfluencies [10]. Finally, prosody may be useful as prior conditioning information for the correct phoneme labeling of an ambiguous acoustic signal. The acoustic implementation of a phoneme depends on its prosodic context in many ways: accented vowels tend to be longer and less subject to coarticulatory variation [2], while accented consonants are produced with greater closure duration [4], greater linguopalatal contact [6], longer voice onset time, and greater burst amplitude [3].

Systems that intend to use prosody only for the purpose of semantic, syntactic, or disfluency processing often implement a prosodic post-processing strategy, in which the input to the prosody recognizer includes a time-aligned word graph generated by an initial prosody-independent speech recognizer. The advantage of a post-processor strategy is greater accuracy, won by the use of syllable-timed acoustic features (e.g., average F0 during the syllable of interest) and word string information [9, 17]. Ostendorf and Ross, for example, perform accent recognition using a stochastic segment recognizer, with segment boundaries given by a hidden Markov model-based (HMM-based) word recognizer [12]. By training and testing a talker-dependent accent recognizer on talker F2B from the Boston Radio News corpus, they are able to achieve 89% correct detection of pitch accent.

The disadvantage of a post-processor strategy is that the front-end recognizer is unable to use prosody to aid in the phonetic labeling of ambiguous acoustic signals. Kompe [9] demonstrates both theoretically

and empirically that a prosody post-processor can improve the search time of a speech recognizer, but never its word recognition accuracy.

Taylor [15] has demonstrated one of the few systems able to recognize pitch accents without prior information about word boundary location. His two-stage prosody recognition system first locates pitch events using an HMM, then labels the pitch events using an analysis-by-synthesis matching strategy. "Pitch events" include high, major pitch accents and rising phrase boundaries. Non-events include minor accents, level accents, and falling boundaries, as well as regions with no perceptible prosodic contour. The best reported pitch event recognition system comprises three-state mixture-Gaussian hidden Markov models of each distinct pitch event label, meaning that every accent type, every boundary type, and every possible combination of an accent and a boundary are distinctly modeled. The HMM observes talker-normalized F0 and delta-F0. For the purposes of scoring recognizer output, all pitch event models (accent and rising boundary models) output the label "e" (event), and all non-event models (including level accent, minor accent, falling boundary, and continuation segment) output the label "c." Event recognition is considered correct only if the overlap between a transcribed pitch event and a true pitch event is at least 50% of the duration of the true event. Under these constraints, speaker-independent pitch event recognition correctness is 72.7%, with a recognition accuracy of 47.7% (25% insertion rate). Speaker-dependent pitch event recognition correctness is 82.1%, with an accuracy of 63.1%.

3 TDRNN Architecture

Here we describe a neural network architecture called time-delay recursive neural network (TDRNN) for automatic recognition of pitch movements. The architecture of the TDRNN is shown in Fig. 1 [8]. The TDRNN is a 3 layer back-propagation network with an additional *long-term context layer*. The TDRNN provides two different representations of dynamic context. First, time-delayed input units (as in a TDNN [16]) allow the representation of short-term context of an explicit trajectory $F0(t)$. Second, multiple recurrent circuits through time-delayed long-term context layer units encode long-term context information that can be used to normalize the input F0 trajectory. The activation of the pitch layer unit at time t is copied into that of the long-term context layer unit which is used for long-term context modelling of pitch movements and acts as an additional input at time $t + 1$.

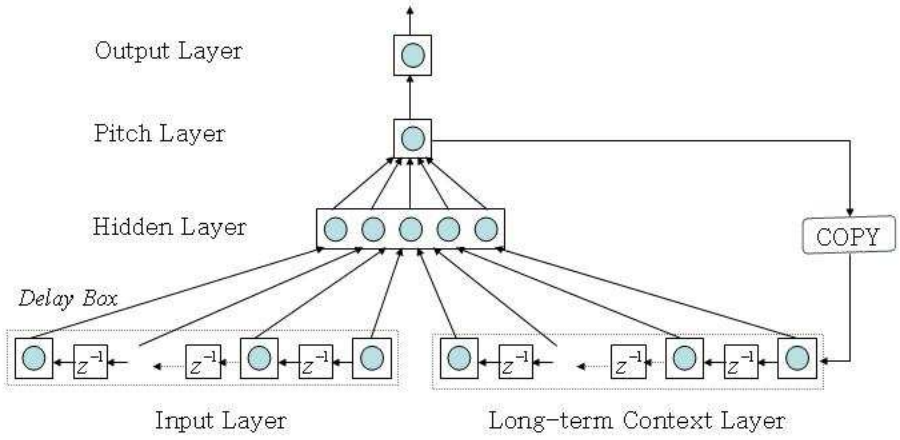

Figure 1: The architecture of TDRNN (Z^{-1} denotes 1 time frame delay).

The *Delay Box* of n interconnections shown in Fig. 1, each with its own time-delay, from the input unit to the hidden unit and between the long-term context unit and the hidden unit can be depicted as Fig. 2. Node i of layer $h-1$ is connected to node j of the next upper layer h, with the connection line having an independent time-delay $\tau_{ijk,h-1}$ and weight $\varpi_{jik,h-1}$. Each node sums up the net inputs from the activation values of the previous layer nodes, through the corresponding time-delay on each connection line, i.e., at time t_n, node j of layer h receives a weighted sum:

$$x_{j,h}(t_n) = f\left(\sum_{i \in N_{h-1}} \sum_{k=1}^{K_{ji,h-1}} \varpi_{jik,h-1} \cdot x_{i,h-1}(t_n - \tau_{ijk,h-1})\right) \tag{1}$$

where $x_{i,h-1}(t_n - \tau_{ijk,h-1})$ is the activation level of layer $h-1$, node i at time $t_n - \tau_{ijk,h-1}$, N_{h-1} denotes the set of nodes of layer $h-1$, $K_{ji,h-1}$ represents the number of connections (i.e., number of time-delays) to node j of layer h from node i of layer $h-1$, and $f(\cdot)$ is a non-decreasing sigmoid function. In [8], both the interconnection weights $\varpi_{jik,h-1}$ and the interconnection delays $\tau_{ijk,h-1}$ were adjusted by an amount proportional to the negative error gradient. In this paper, however, only the interconnection weights $\varpi_{jik,h-1}$ are learned using the error back-propagation learning algorithm [5], and the delays are fixed. Since the activation of the recurrent node is a direct copy of the previous pitch layer activation, the feedback

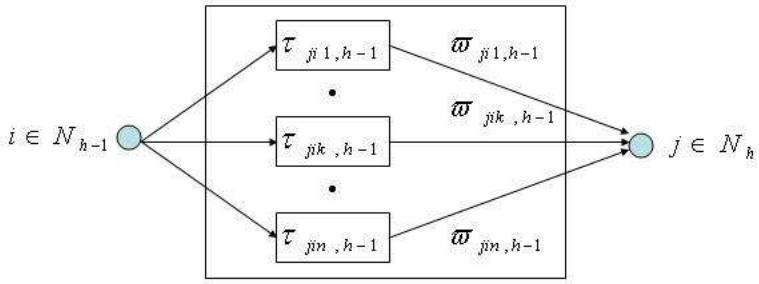

Figure 2: Delay box ($\tau_{ijk,h-1}$ denotes kth time frame delay to node j from node i).

connection through the *copy* operation is not subject to training.

4 Experimental Methods

The TDRNN was trained and tested for the purpose of talker-independent, gender-dependent pitch event recognition using data extracted from the Boston Radio News Corpus [11]. Performance of the TDRNN was compared to the performance of a TDNN/MLP (time-delay neural network/multi-layer perceptron) and an HMM-based recognizer trained and tested on the same task.

The Boston Radio News Corpus is a series of radio stories read by seven professional radio announcers, and partially annotated using the ToBI (tones and break indices) prosodic annotation system [13]. Seven types of pitch accents are transcribed in the Radio News Corpus. All seven types of accents involve classification of pitch movement on the accented syllable into one of three categories: high (H*), downstepped (!H*), and low (L*). The notation "*?" is used to mark a questionable pitch accent. Some transcriptions mark the location of a pitch accent (as "*") but not its type; most of these are high or downstepped accents. The TDRNN, MLP, and HMM recognizers in our experiments are trained to recognize as *pitch events* all syllables marked with H*, !H*, *, ?*, or L*, and as *non-events* all unaccented syllables. For training purposes, each pitch *event* or *non-event* starts at the beginning of the first sonorant phoneme in a syllable, and ends at the end of the last sonorant phoneme in the same syllable. For testing purposes, all three recognizers were used to label every frame in the test database as either accented or unaccented. During recognition tests, a pitch event was

considered correctly recognized if at least 50% of its frames were labeled "accented." The TDRNN, MLP, and HMM are trained using 67 paragraphs comprising 2,078 pitch events and 2,116 non-events from one female speaker (F1A), and tested with 164 paragraphs comprising 6,999 pitch events and 7,082 non-events from another female speaker (F2B).

The TDRNN and MLP recognizers observe a two-dimensional input vector containing normalized fundamental frequency ($\tilde{F}_0(t)$) and energy ($\tilde{E}_0(t)$). The fundamental frequency $F_0(t)$ is extracted using the `formant` program in Entropic XWAVES with probability of voicing (PV) output as a confidence measure for the extracted $F_0(t)$. We eliminated pitch doubling and halving errors by eliminating $F0$ that falls into the doubling and halving clusters of a 3 mixture Gaussian model whose mixture component means are restricted to $1/2\mu$, μ, and 2μ, where μ is the estimated utterance mean F_0. We then normalize F_0 by μ and convert it to log scale:

$$\hat{F}_0(t) = \max\left(0.2, \log\left(\frac{F_0(t)}{\mu} + 1\right)\right). \tag{2}$$

To eliminate unreliable $\hat{F}_0$ measures, those with PVs smaller than a heuristic threshold are replaced by the linear interpolated values $\tilde{F}_0$ based on the $\hat{F}_0$ that have PVs greater than the threshold. Similarly, energy is normalized using:

$$\tilde{E}_0(t) = \max\left(-3, \log\left(\frac{E0(t)}{\eta}\right)\right), \tag{3}$$

where η is the utterance maximum energy.

The TDRNN is configured with 2 input units, 5 hidden units, 1 pitch layer unit (1 long-term context layer unit), and 1 output unit. The input units have 2 time frame delays for input context modelling, while the long-term context layer unit has 18 time frame delays for the long-term context modelling of pitch movements. The MLP is configured with 2 input units, 10 hidden units, and 2 output units. The input units have 10 time frame delays to provide context to the network, and 11 frames are used as input. The HMM-based recognizer uses five three-state HMMs, modeling the five labels H*, !H*, ?*, L*, and unaccented (there were no * labels in the training data). Of several tested HMM configurations, best performance was achieved using a ten-component diagonal-covariance mixture Gaussian PDF with a six-dimensional feature vector comprising $\tilde{F}_0(t)$, $\tilde{E}_0(t)$, and their deltas and delta-deltas. The HMMs have 393 trainable parameters

each, for a total of 2358 trainable parameters. The MLP has a total of 240 trainable parameters (weights), and the TDRNN has 126 trainable parameters.

The TDRNN is trained, using error back-propagation, to imitate a target function. The target function for the TDRNN is equal to 1 during pitch events, and 0 otherwise (thus the TDRNN target function is equal to 0 during pitch non-events, and also during non-sonorant frames). The two output units of the MLP are trained, using error back-propagation, to imitate complementary target functions: one is equal to the target function of the TDRNN, while the other is equal to one minus the TDRNN target. The HMMs are trained using a standard Baum-Welch maximum likelihood training algorithm.

5 Experimental Results

Fig. 3 shows three functions of time, computed for a sentence in the test database: the TDRNN target function, the TDRNN output layer unit, and the TDRNN pitch layer unit. As shown, the TDRNN output layer unit tracks the target function reasonably well. The figure also demonstrates the way in which the pitch layer unit encodes information about the long-term context of the input pitch contours. Specifically, the activation of the pitch layer unit approximates, reasonably well, the complement of the target function: the probability of a pitch non-event.

In recognition experiments, the MLP marks each frame as accented if and only if the activation of the pitch-event output node is higher than that of the non-event node. The TDRNN marks each frame as accented if and only if the activation of the output layer unit is higher than that of the pitch layer unit. Pitch event recognition accuracy is computed as the number of correctly recognized events divided by the number of events in the database; in order to be correctly recognized, at least 50% of the frames in an event must be marked as "accented." Non-event recognition accuracy is scored the same way.

Table 1 summarizes pitch event and non-event recognition results on the test data. The TDRNN shows the correct recognition of 91.9% of pitch events and 91.0% of pitch non-events, for an average accuracy of 91.5% over both pitch events and non-events. The MLP with contextual input exhibits 85.8%, 85.5%, and 85.6% recognition accuracy respectively, while the HMMs show the correct recognition of 36.8% of pitch events and 87.3% of pitch non-events, for an average accuracy of 62.2% over both pitch events and non-events. These results suggest that the TDRNN encodes dynamic variations of pitch movements better than

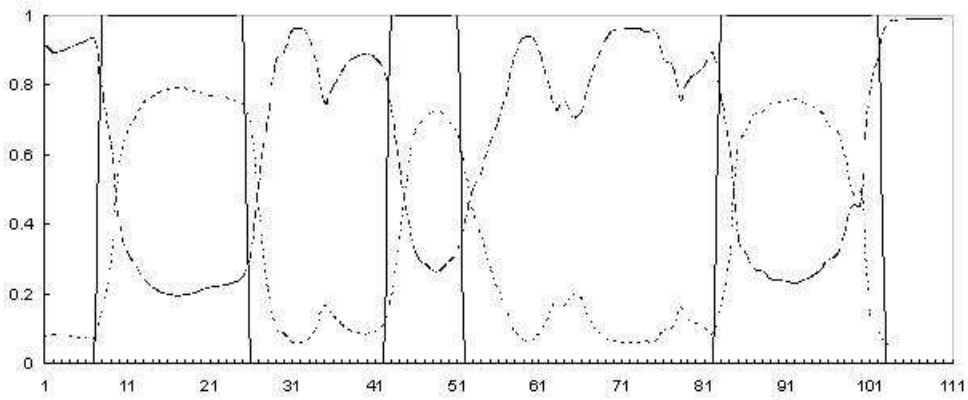

Figure 3: The solid line is for true pitch event, the dotted line is for the output values of the output layer unit of TDRNN, and the dashed line is for the output values of the pitch layer unit which estimates pitch non-events.

the MLP and HMM do. The HMM, in particular, appears to be strongly biased in favor of a "non-event" label, possibly because maximum likelihood training results in suboptimal selection of the decision boundary between these two categories.

6 Conclusions

Both the MLP and the TDRNN architecture provide significantly better performance than a mixture-Gaussian-based HMM recognizer (85.6% and 91.5% versus 62.2%). The word-independent TDRNN recog-

	TDRNN		MLP		HMM	
ToBI Label	"Event" Percent (N)	"Non-Event" Percent (N)	"Event" Percent (N)	"Non-Event" Percent (N)	"Event" Percent (N)	"Non-Event" Percent (N)
H*	93.0% (4331)	7.0% (325)	87.8% (4088)	12.2% (568)	37.2% (1733)	62.8% (2923)
!H*	91.5 (1392)	8.5 (130)	87.3 (1329)	12.7 (193)	36.9 (562)	63.1 (960)
*	94.7 (126)	5.3 (7)	94.0 (125)	6.0 (8)	44.4 (59)	55.6 (74)
?*	88.7 (337)	11.3 (43)	75.8 (288)	24.2 (92)	36.8 (140)	63.2 (240)
L*	78.9 (243)	21.1 (65)	57.5 (177)	42.5 (131)	26.9 (83)	73.1 (225)
All Accents	91.9 (6429)	8.1 (570)	85.8 (6007)	14.2 (992)	36.8 (2577)	63.2 (4422)
Unaccented	9.0 (634)	91.0 (6448)	14.5 (1030)	85.5 (6052)	12.7 (897)	87.3 (6185)

Table 1: Confusion matrices, showing, for each ToBI label, the percentage of tokens (number of tokens in parentheses) recognized as "pitch event" or "non-event" by the TDRNN, MLP, and HMM classifiers.

nizer also outperforms the best previously published word-independent accent recognition results (63.1% [15]) as well as the best published word-segmentation-dependent accent classification results using the Radio News Corpus (89% [12]). Both of these previous systems, like our HMM system, are mixture-Gaussian-based Bayesian classifiers trained using a maximum likelihood criterion, while the MLP and the TDRNN architecture use discriminatively trained, non-parametric classification models. Performance of a parametric classifier depends on the details of the probability model in the critical region near the classification threshold; it is possible that the probability distribution of $\tilde{F}_0(t)$ and $\tilde{E}_0(t)$ near the decision boundary is poorly approximated by a mixture Gaussian model trained using the maximum likelihood method.

The recognition accuracy of the talker-independent TDRNN architecture proposed in this paper is almost identical to the rate of agreement among human transcribers (91.5% versus 91%) [11]. The TDRNN performs significantly better than a non-recursive MLP architecture (85.6%). These results suggest that long-term context modeling through multiple recurrent circuits is useful for the correct recognition of pitch events.

7 Acknowledgments

This work was supported in part by NSF award number 0132900, and in part by a grant from the University of Illinois Critical Research Initiative. Statements in this paper reflect the opinions and conclusions of the authors, and are not endorsed by the NSF or the University of Illinois.

References

[1] Mary E. Beckman and Janet Pierrehumbert. Intonational structure in Japanese and English. *Phonology Yearbook*, 3:255–309, 1986.

[2] T. Cho. *Effects of Prosody on Articulation in English*. PhD thesis, UCLA, 2001.

[3] Jennifer Cole, Hansook Choi, Heejin Kim, and Mark Hasegawa-Johnson. The effect of accent on the acoustic cues to stop voicing in radio news speech. In *Proc. Internat. Conf. Phonetic Sciences*, 2003.

[4] Kenneth DeJong. The supraglottal articulation of prominence in English: Linguistic stress as localized hyperarticulation. *J. Acoust. Soc. Am*, 89(1):369–382, 1995.

[5] Rumelhart D. E., McClelland J. L., and the PDP Research Group. Learning representations by back-propagating errors. In *Paralell Distributed Processing*, volume 1, pages 318–362. MIT Press, 1986.

[6] Cecile Fougeron and Patricia Keating. Articulatory strengthening at edges of prosodic domains. *J. Acoust. Soc. Am*, 101(6):3728–3740, 1997.

[7] Ji-Hwan Kim and Philip C. Woodland. The use of prosody in a combined system for punctuation generation and speech recognition. In *Proc. EUROSPEECH*, 2001.

[8] Sung-Suk Kim. Time-delay recurrent neural network for temporal correlations and prediction. *Neurocomputing*, 20:253–263, 1998.

[9] R. Kompe. *Prosody in Speech Understanding Systems*. Springer-Verlag, 1997.

[10] Christine H. Nakatani and Julia Hirschberg. A corpus-based study of repair cues in spontaneous speech. *J. Acoust. Soc. Am*, 95(3):1603–1616, 1994.

[11] M. Ostendorf, P.J. Price, and S. Shattuck-Hufnagel. *The Boston University Radio News Corpus*. Linguistic Data Consortium, 1995.

[12] M. Ostendorf and K. Ross. A multi-level model for recognition of intonation labels. In *Computing prosody: computational models for processing spontaneous speech*. Springer-Verlag New York, Inc., 1997.

[13] Joseph F. Pitrelli, Mary Beckman, and Julia Hirschberg. Evaluation of prosodic transcription labeling reliability in the TOBI framework. In *Proc. ICSLP*, 1994.

[14] P.J. Price, M. Ostendorf, S. Shattuck-Hufnagel, and C. Fong. The use of prosody in syntactic disambiguation. *J. Acoust. Soc. Am*, 90(6):2956–2970, Dec. 1991.

[15] Paul Taylor. Analysis and synthesis of intonation using the Tilt model. *J. Acoust. Soc. Am*, 107(3):1697–1714, 2000.

[16] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Trans. ASSP*, 37:328–339, 1989.

[17] Colin Wightman and Mari Ostendorf. Automatic labeling of prosodic patterns. *IEEE Trans. Speech and Audio Processing*, 2(4):469–481, Oct. 1994.