Using Deep Neural Networks for Smoothing Pitch Profiles in Connected Speech

Michele Ferro*
CELI Language Technology

Fabio Tamburini**
Università di Bologna

This paper presents a new pitch tracking smoother based on deep neural networks (DNN). It leverages Long Short-Term Memories, a particular kind of recurrent neural network, for correcting pitch detection errors produced by state-of-the-art Pitch Detection Algorithms. The proposed system has been extensively tested using two reference benchmarks for English and exhibited very good performances in correcting pitch detection algorithms outputs when compared with the gold standard obtained with laryngographs.

1. Introduction

The pitch, and in particular the fundamental frequency - F0 - which represents its physical counterpart, is one of the most relevant perceptual parameters of the spoken language and one of the fundamental phenomena to be carefully considered when analysing linguistic data at a phonetic and phonological level. As a consequence, the automatic extraction of F0 has been a subject of study for a long time inspiring many works that aim to develop algorithms, commonly known as Pitch Detection Algorithms (PDA), able to reliably extract F0 from the acoustic component of the utterances.

Technically, the extraction of F0 is a problem far from trivial and the great variety of methodologies applied to this task demonstrate its extreme complexity, especially considering that it is difficult to design a PDA that works optimally for the different recording conditions, considering that parameters such as speech type, noise, overlaps, etc. are able to heavily influence the performances of this kind of algorithms.

Scholars worked hard searching for increasingly sophisticated techniques for these specific cases, although extremely relevant for the construction of real applications, considering solved, or perhaps simply abandoning, the problem of the F0 extraction for the so-called "clean speech". However, anyone who has used the most common programs available for the automatic extraction of F0 is well aware that errors of halving or doubling of the value of F0, to cite only one type of problem, are rather common and that the automatic identification of voiced areas within the utterance still poses numerous problems.

Every work that proposes a new method for the automatic extraction of F0 should accomplish an evaluation of the performances obtained in relation to other PDAs, but, usually, these assessments suffer from the typical shortcomings deriving from evaluation systems: they usually examine a very limited set of algorithms, often not available in their implementation, typically considering corpora not distributed, related to specific languages and/or that contain particular typologies of spoken language (pathological, disturbed by noise, overlapped dialogues, singing voices, etc.) (Veprek and Scordilis 2002; Wu, Wang, and Brown 2003; Kotnik, Höge, and Kacic

* CELI, Via S. Quintino, 31, 10121 Torino, Italy. E-mail: lele.ferro4@gmail.com
** Dept. of Classic Philology and Italian Studies, Via Zamboni 32, 40126 Bologna, Italy. E-mail: fabio.tamburini@unibo.it

2006; Jang et al. 2007; Luengo et al. 2007; Chu and Alwan 2009; Bartosek 2010; Huang and Lee 2012; Chu and Alwan 2012; Babacan et al. 2013; Gawlik and Wszolek 2018). There are a few studies, among the most recent, that have performed quite complete evaluations that are based on standard speech corpora often freely downloadable (de Cheveigné and Kawahara 2002; Camacho 2007; Wang and Loizou 2012; Sukhostat and Imamverdiyev 2015; Jouvet and Laprie 2017). Most research works use a single metric in the assessment that measures a single type of error, not considering or partly considering the whole panorama of indicators developed from the pioneering work of Rabiner and colleagues (1976) and therefore, in our opinion, the results obtained seem to be rather partial.

Tamburini (2013) performed an in-depth study of the different performances exhibited by several widely used PDAs by using standard evaluation metrics and well-established corpus benchmarks.

Starting from this study, the main purpose of our research was to improve the performances of the best Pitch Detection Algorithms identified in (Tamburini 2013) by introducing a post-processing smoother. In particular, we implemented a pitch smoother adopting Keras[1], a powerful high-level neural networks Application Program Interface (API), written in Python and able to run on top of TensorFlow, one of the most powerful machine learning libraries especially devoted to the development of large neural network models.

The paper is organised as follows: in Section 2 we will describe the pitch smoothing problem; in Section 3 we will present our neural PDA smoother while in section 4 we will define the experiments we did to evaluate our proposal; Section 5 shows the results and in Section 6 we will draw some provisional conclusions and propose some future works.

2. Pitch error correction and smoothing

Typical PDAs are organised into two different modules: the first stage tries to detect pitch frequencies frame by frame and, in the second stage, the pitch candidates, along with their probabilities, are connected into pitch contours using dynamic programming techniques (Bagshaw 1994; Chu and Alwan 2012; Gonzalez and Brookes 2014) or hidden Markov models (HMMs) (Jin and Wang 2011; Wu, Wang, and Brown 2003). In this second stage, the different PDAs apply various techniques in order to correct the intonation profile removing various errors produced by the first step.

These techniques are, however, not completely satisfactory and various types of errors remain in the intonation profile. That is why in the literature we can find several studies aiming at proposing pitch profile smoothers that further post-process the PDAs output trying to enhance the profile correctness. Some works try to correct intonation profiles by applying traditional techniques (Zhao, O'Shaughnessy, and Minh-Quang 2007; So, Jia, and Cai 2012; Jlassi, Bouzid, and Ellouze 2016), while few others (see for example (Kellman and Morgan 2017; Han and Wang 2014)) are based on DNN (either Multi-Layer Perceptrons or Elman Recurrent Neural Networks).

A complex periodic sound will actually have multiple repeating patterns in its waveform: some repeating at faster rates and some taking longer to repeat their cycles. It is the slowest (the longest period/lowest frequency) repeating pattern in a complex periodic sounds that governs the signal's perceived pitch. It is important mentioning the difference between perceptual and quantitative properties. Starting from this contrast, the pitch of a sound can be defined as the mental sensation or perceptual correlate of fundamental frequency; in general, if a sound has a higher fundamental frequency we perceive it as having a higher pitch. The relationship is not

1 https://keras.io/

linear, since human hearing has different responses for different frequencies. Roughly speaking, human pitch perception is most accurate between 100 Hz and 1000 Hz, and in this range pitch correlates linearly with frequency. Human hearing represents frequencies above 1000 Hz less accurately and above this range pitch correlates logarithmically with frequency.

F0 can be seen as the minimum frequency of the vocal folds vibration, or the frequency of the complex wave. All complex periodic sounds or waves can be mathematically analyzed as being composed of multiple single-frequency sounds/waves, such a series of sine waves: the Fourier's theorem states that any periodic signal is composed of the summation of multiple sine waves with particular amplitudes and phases. Fourier's theorem by extension implies that we can decompose complex periodic sounds into simple components (Byrd and H.Mintz 2010). The frequencies of a signal's harmonics are integer multiples of its F0: for this reason the second harmonic is 2 x F0, the third harmonic is 3 x F0 and so on. We cannot tell simply by looking at a complex waveform what its component frequencies or harmonics are. A computer is generally used to implement algorithms based on Fourier's theorem to find a complex signal's harmonics. A different kind of display, called a power spectrum, can be useful for showing the frequency composition or spectrum of a sound frame. A power spectrum, like in Figure 1, plots frequency on the horizontal axis and amplitude (or magnitude) on the vertical axis.

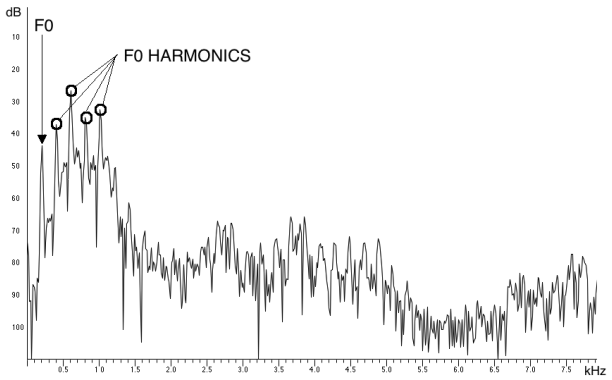

Figure 1
Power spectrum of a speech sample frame showing F0 and its harmonics.

Despite the number of studies devoted to the design of efficient PDAs, correct pitch extraction remains an open problem for various reasons. Pitch estimation, indeed, is a process heavily influenced by phenomena observed in spontaneous speech:

- F0 varies in time, potentially at each period of vibration of vocal folds;
- it often happens that "true" F0 has sub-harmonics as its submultiples which alter estimation of values in contrast with perception;
- the presence of resonances and filters in the vocal tract can emphasize harmonics of F0 multiples of the real value;
- sonority is often very irregular at the beginning and at the end of a voiced linguistic segment and all the frames involved in these transitions have minimal similarities between the corresponding waveforms;
- even for human experts the classification of the boundaries of voiced areas is a far from an easy task;

- due to certain disturbances it is possible that signals occur with a relevant percentage of periodicity in unvoiced areas too;
- voiced regions have a wide dynamic range of amplitude;
- it is difficult to distinguish periodic background noise from breathy voice;
- some voiced intervals are very short and they can be composed of just two or three cycles.

These different and complex problems have determined the spread of studies about F0 detection. We will focus on some of these algorithms later in this contribution.

The range of fundamental frequencies found in human voices is roughly 60 to 500 Hz, but in adult males a typical F0 might be 120 Hz; in a female voice a typical F0 might be 225 Hz, and in a child it might be 265 Hz. It is worth underlying that variation in fundamental frequency in speech is due to the structure of the larynx and the vocal folds only (Byrd and H.Mintz 2010). In addition to voicing, there are many ways to generate noise or sources of sound in the vocal tract during speech. For example, a fricative consonant creates noise by the turbulent airflow generated when air is forced through a narrow constriction, sometimes directed against the teeth as an obstacle. In this case, unlike the voicing source, the acoustic energy is generated in the mouth, not at the larynx. We state this because it has to be understood that many sound sources occur in speech, such as the noise created when a stop constriction is opened, but we will concentrate on the main sound source in speech - the voicing source - and look next at how the harmonic structure of this source is shaped by the vocal tract.

Here, we will focus on a specific category of pitch detection errors, the halving and doubling errors, in which the fundamental frequency F0 is confused with one of its harmonics (or sub-harmonics), generating incorrect assignments to multiple (or sub-multiple) frequency values of the correct one (Murray 2001). More precisely, F0 doubling errors occur when the estimated fundamental frequency is an overtone of the real fundamental frequency; on the other hand, F0 halving errors occur when the F0 determination algorithm erroneously mistakes the correctly estimated fundamental frequency by dividing the correct F0 value by some multiple of two. The most sophisticated algorithms tend to apply appropriate post-processing procedures in order to properly identify the correct value, among several possible candidates typically ranked in some way by the F0 extraction algorithm.

We will return to our brief description of halving and doubling errors later in this paragraph; now we provide a description of the smoothing method proposed by (Bagshaw 1994) in order to clarify the problem. The main purpose of this procedure is to distinguish between legitimate variations in the pitch profile and errors, trying to correct these in the best way. In particular, there is the assumption that F0 can grow between a frame and the next one to the maximum of the 75% and consequently it can drop to the 25% of the value of the first of the two frames. All values outside this range are considered respectively doubling and halving errors. At this point each voiced section of the utterance is processed separately: all the F0 values in the different frames which make up the voiced area are divided in various groups, each of them denoted by an index between -2 and 2. The partition begins putting the F0 values in the group identified with the index 0 as long as the transition among two subsequent frames generates a potential halving or doubling error. If this happens, the following F0 values are respectively positioned in the group identified with the index -1 or 1. The procedure continues in this way until all the values in the voiced region are placed in a group, changing the index of the group each time a potential error is detected. When the operation of subdivision of F0 values in each group ends, the procedure of correction of halving and doubling errors begins: the group containing the largest quantity of values is identified, defining it as the condition of normality (it could be the

group indexed with 0 or even a different group). Then, groups with a higher index are considered containing doubling errors while groups with a lower index are considered containing halving errors. Consequently the entire set of errors is corrected multiplying and dividing by powers of 2 the F0 values collected in the groups that identify incorrect estimates of the real fundamental frequency value.

We report also the research carried out by (Brøndsted 1997) according to which for a specific dialect of Danish, the presence of a glottal consonant "stød" can cause a pitch tracker to incorrectly report a halved value, as an example of a pitch tracking problem intimately connected with specific phonetic configurations. A further step would be to coordinate descriptions of pitch tracking doubling and halving errors with respect to categorizations of laryngealization (sometimes called creaky voice). This is a special kind of phonation in which the arytenoid cartilages in the larynx are drawn together; as a result, the vocal folds are compressed rather tightly, becoming relatively slack and compact. They normally vibrate irregularly at 20-50 pulses per second, about two octaves below the frequency of normal voicing, and the airflow through the glottis is very slow. Although creaky voice may occur with very low pitch, as at the end of a long intonation unit, it can also occur with a higher pitch (Titze 1994). The phenomenon of laryngealization is involved in the context of "cut-off" words, for example those words that a speaker does not complete (Shriberg 1999).

A better recognition of glottal pulses may lead to improve cut-off words recognition which are difficult phenomena to determine for a pitch tracker and consequently for an Automatic Speech Recognition (ASR) system too. Regarding this aspect, one can opt for an harmonic model of speech, which has gained considerable attention recently. This model takes into account the harmonic nature of voiced speech and it can be formulated to estimate pitch candidates with maximum likelihood criterion. Without entering deeply into the matter, the popular source-channel model of voiced speech considers glottal pulses as a source of period waveforms which is being modified by the shape of the mouth assumed to be a linear channel. Thus, the resulting speech is rich in harmonics of the glottal pulse period (Stylianou 1996). Like in other PDAs, pitch doubling and halving errors affect the harmonic model too; in order to solve these problems, one can opt for a local smoothing function that exploits the fact that there is more energy in the harmonics near the true pitch than at the corresponding neighbourhoods of half or double of its value. A local smoothing function is employed to include this energy and improve the strength of the pitch candidates in each frame. The harmonic model requires specification of the number of harmonics and the optimal choice depends on noise conditions (Asgari and Shafran 2013).

Here we provided a brief analysis of doubling and halving errors, a description of a procedure of pitch smoothing, some language dependent problems and the employment of the harmonic model to solve some of them. Starting from the next section we put our attention on our own proposal.

3. A Neural PDA smoother

The main purpose of our research work was an attempt to improve the performances of the Pitch Detection Algorithms. It is relevant to underline that all PDAs embody, as a last stage, some kind of smoothing algorithm trying to capture and correct mistakes in the intonation profile. As discussed before, these methods are often not sufficient to provide a reliable contour throughout the whole utterance. The Neural Smoother we are proposing tries to further improve profile smoothing applying more powerful techniques.

Our first assumption regarded the typology of the artificial neural network to employ. In order to correct the PDAs results, our pitch smoother needed to operate an increasingly precise approximation from the pitch input sequence to be improved to the gold standard output target obtained from the laryngograph. Having configured our problem as a sequence-to-sequence

mapping, we employed a particular architecture of recurrent neural network (RNN) suitable for this kind of problem.

These networks are recurrent because they perform the same computations for all the elements of a sequence of inputs, and the output of each element depends, in addition to the current input, from the previous state. RNNs have proved to have excellent performances in problems such as predicting the next character in a text or, similarly, the prediction of the next word in a sentence. They are also used for more complex problems, such as Machine Translation and Text Summarisation. In the former case, the network gets as input a sequence of words in a source language, while the output will be translated from the input sequence in a target language. Finally, other applications of great importance in which the RNNs are widely used are speech recognition and also image recognition.

A Long Short Term Memory (LSTM) neural net is a special Recurrent Neural Network architecture that was originally conceived by (Hochreiter and Schmidhuber 1997). This kind of neural network has gained a lot of attention in the context of deep learning because it offers excellent results and performances. The LSTM based networks are ideal for temporal sequences prediction and classification, replacing many traditional approaches to deep learning.

LSTM is a network composed by cells (LSTM blocks) linked to each other. Each LSTM block contains three types of gate: Input gate, Output gate, and Forget gate, which broadly implement, respectively, the function of writing, reading, and resetting on the cell memory. More precisely, the Input gate regulates the possibility for a new value to enter into the cell, the Forget gate determines if the value will be retained into the cell or not and the Output gate controls to which extent the cell value is transferred into the block output. Some of the connections between the LSTM elements are recurrent and all the weights of the connections have to be learned during the training process. The presence of these gates allows LSTM cells to remember information for a long time reducing the problem of the vanishing/exploding gradients during the training.

Mathematically, we can formalise the behaviour of a standard LSTM cell as

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
\tag{1}
$$

where $x_t \in \mathbb{R}^d$ is input vector to the LSTM unit, $f_t \in \mathbb{R}^h$ the input gate's activation vector, $o_t \in \mathbb{R}^h$ the output gate's activation vector, $h_t \in \mathbb{R}^h$ the hidden state vector also known as output vector of the LSTM unit, $c_t \in \mathbb{R}^h$ the cell state vector, $W \in \mathbb{R}^{h \times d}$, $U \in \mathbb{R}^{h \times h}$, $b \in \mathbb{R}^h$ the weight matrices and bias vectors parameters which need to be learned during training, $\sigma_g, \sigma_h, \sigma_c$ the activation functions and the superscripts d and h refer to the number of input features and to the number of hidden units, respectively.

More specifically, in our case study we decided to employ a bidirectional LSTM. Bidirectional neural networks are based on the idea that the output at time t may depend on previous and future elements in the sequence. To realize this, the output of two neural networks must be mixed: one executes the process in one direction and the second in the opposite direction by processing the reversed input sequence. The network splits neurons of a normal recurrent neural network into two directions, one for positive time verse (forward states), and another for negative time verse (backward states) concatenating the outputs of the two networks. By this structure, the output

layer can get information from past and future states. We decided to opt for bidirectional LSTMs in order to have a better performance in our sequence learning (or approximation) problem.

We decided also to one-hot encode all the frames of our sequences, in order to obtain better performances in our sequence learning task. For our specific case, since we were working on female e male sources in both our datasets, we chose an interval of [0, 499] Hz for the number of features. Therefore, we transformed the F0 values determined for each frame in order to obtain input/output one-hot vectors; on the other hand, for the final evaluation of the predictions made by our model, we reversed this transformation getting common pitch values in the interval [0-499] Hz. This encoding of input and output data leads to input/output vectors of size 500 in our neural network model.

4. Experiments

4.1 Neural PDA setup

We implemented our pitch smoother in Python adopting Keras and Tensorflow. We defined a bi-directional Long Short Term Memory neural network layer with 100 neurons for one direction of the sequence and 100 neurons for the other direction, with a total of 200 LSTM units. A TimeDistributed layer has been wrapped around the output layer so that one value per timestep could be predicted given the full sequence provided as input. This allowed the LSTM hidden layer to return a sequence of values (one per timestep) rather than a single value for the whole input sequence. The network was optimised by using the categorical cross entropy loss function and the Adam optimiser algorithm (Kingma and Ba 2015).

4.2 Tested PDAs

We chose to test the three PDAs exhibiting the best performances in (Tamburini 2013), namely RAPT, SWIPE' and YAAPT. Even though they were originally developed as MATLAB functions, we decided to adopt the corresponding Python implementations and thus, as a first step, we have to test the correspondence of performances of the python implementations with the original ones in MATLAB.

4.2.1 A Robust Algorithm for Pitch Tracking (RAPT)

The primary purpose in the development of RAPT (Talkin 1995) was to obtain the most robust and accurate estimates possible, with little thought to computational complexity, memory requirements or inherent processing delay. This PDA was designed to work at any sampling frequency and frame rate over a wide range of possible F0, speaker and noise condition. In fact, although the delay inherent in RAPT probably disqualifies it from use in standard telephony, it does operate continuously and can be used anywhere. About this matter, several efficiency enhancements have been incorporated that significantly reduce computational costs while maintaining the desired accuracy. More specifically, for the determination of the pitch profile, RAPT adopts a Normalized Cross-Correlation Function (NCCF) and each candidate of F0 is estimated thanks to dynamic programming (also known as dynamic optimization, a method employed for solving a complex problem by breaking it down into a collection of simpler subproblems). The Python implementation we used is available at `http://sp-tk.sourceforge.net/`.

4.2.2 The Sawtooth Waveform Inspired Pitch Estimator (SWIPE/SWIPE')

SWIPE (Camacho 2007) improves the performance of pitch tracking adopting these measures: it avoids the use of the logarithm of the spectrum, it applies a monotonically decaying weight

to the harmonics, then the spectrum in the neighbourhood of the harmonics and middle points between harmonics are observed and smooth weighting functions are used. We will not focus on an overview of the mathematical expression of this PDA, but, in general, the algorithm can be described as the computation of the similarity between the square-root of the spectrum of the signal and the square-root of the spectrum of a sawtooth waveform, using a pitch dependent optimal window size. This definition gave rise to the name Sawtooth-Waveform Inspired Pitch Estimator (Camacho 2007). In our research we adopted SWIPE', a variant of this PDA that adopts only the main harmonics for pitch estimation, implemented in Python and available again at `http://sp-tk.sourceforge.net/`.

4.2.3 Yet Another Algorithm for Pitch Tracking (YAAPT)

YAAPT (Zahorian and Hu 2008) is a fundamental frequency (Pitch) tracking algorithm which was designed to be highly accurate and very robust for both high quality and telephone speech. One of the key features of YAAPT is the usage of spectral information to guide F0 tracking. Spectral F0 tracks can be derived by using the spectral peaks which occur at the fundamental frequency and its harmonics. It is experimentally shown that the F0 track obtained from the spectrogram is useful for refining the F0 candidates estimated from the acoustic waveform, especially in the case of noisy telephone speech (Zahorian and Hu 2008). With relation to the functioning of this PDA, a preprocessing step is employed to create multiple versions of the signal. Consequently, spectral harmonics correlation techniques (SHC) and a Normalized Cross-Correlation Function (as in RAPT) are adopted. The final profile of F0 is estimated thanks to dynamic programming techniques. For our experiments we employed pYAAPT, a Python implementation available at `http://bjbschmitt.github.io/AMFM_decompy/pYAAPT.html`.

4.3 Gold Standards

The evaluation tests were based on two English corpora considered as gold standards, both freely available and widely used in literature for the evaluation of PDAs:

- Keele Pitch Database - KPD[2] (Plante, Meyer, and Ainsworth 1995): it is composed of 10 speakers, 5 males and 5 females, who read, in a controlled environment, a small phonetically balanced text (the 'North Wind story'). The corpus contains also the output of a laryngograph, from which it is possible to accurately estimate the value of F0.
- FDA[3] (Bagshaw, Hiller, and Jack 1993): it is a small corpus containing 5' of recordings divided into 100 utterances, read by two speakers, a male and a female, particularly rich in fricative sound, nasal, liquid and glide, sounds particularly problematic to be analysed by the PDAs. Also in this case the gold standard for the values of F0 is estimated starting from the output of the laryngograph.

It is worth noticing that each of these datasets contains the output of a laryngograph. This instrument is composed of a pair of disc electrodes to record the vibrations around the throat. Electroglottography (EGG) signals record the time varying displacement of air particles at the glottis during the production of voiced sounds such as vowels, semi-vowels, nasals, diphthongs and voiced consonants. The electrodes are placed, non-invasively, at either side of the larynx.

2 `https://lost-contact.mit.edu/afs/nada.kth.se/dept/tmh/corpora/KeelePitchDB/`
3 `http://www.cstr.ed.ac.uk/research/projects/fda/`

A high-frequency electric current is applied, and due to variance in electrical impedance from the opening and closing of the glottis, an electroglottogram can be produced. There are several advantages of using EGG, the most significant being to reduce background noise. By eliminating irrelevant signals, EGG can increase the accuracy in the identification of perceived pitch. In the future, real-life applications of EGG can be developed due to its ability to reduce background noise, such as a wireless EGG integrated with clothes (Hui et al. 2015). This fact had crucial implications for the aims of our contribution: using KPD we encountered a few problems due to corrupted data. As (Plante, Meyer, and Ainsworth 1995) pointed out, where they knew that there was voiced speech but the larynx trace was corrupted, the data have been set to -1 (this happened sometimes because the measurements were based on two electrodes on the skin, which could lose contact as the speakers moved around). We will explain later how we decided to treat these corrupted data.

To perform our experiments, we had to split our datasets into a training set, a validation set and a test set. Consequently, we trained our model on the training set, we used the validation set to tune the hyperparameters of our smoother and finally the test dataset was employed to provide a balanced evaluation of our final model. This procedure was adopted both on KPD and FDA files, considering the output sequences of our PDAs and the gold standards obtained from the laryngograph. The main differences among the two datasets were the total number of files (10 speech samples for KPD and 100 for FDA) and the size of the files themselves. In fact, the original KPD files were much bigger than those of FDA, thus we decided to split each of them into 4 slices obtaining 40 speech samples. Considering that our purpose was trying to correct the sequences of the output of RAPT, pYAAPT and SWIPE' PDAs, we had in total 6 experiments (3 PDAs x 2 datasets).

In order to operate a significant subdivision between female and male files, we present our splitting for Keele Pitch Database:

	Training set	**Validation set**	**Test set**
Females	12	4	4
males	12	4	4

Here, instead, the splitting for FDA:

	Training set	**Validation set**	**Test set**
Females	34	8	8
males	34	8	8

We considered also the possibility of joining the two datasets in order to see if we get some improvements (Mixed configuration), and we followed the splitting

	Training set	**Validation set**	**Test set**
Females	46	12	12
males	46	12	12

All the splittings are speaker based as the speakers in the validation and test sets are not part of the training set.

4.4 Evaluation metrics

Proper evaluation mechanisms have to introduce suitable quantitative measures of performance that should be able to grasp the different critical aspects of the problem under examination. In (Rabiner et al. 1976) a de facto standard for PDA assessment measures is established, a standard used by many others after him (e.g. (Chu and Alwan 2009)). Given $E_{voi \to unv}$ and $E_{unv \to voi}$, respectively representing the number of frames erroneously classified between voiced

and unvoiced and vice versa, and E_{f0}, denoting the number of voiced frames in which the pitch value produced by the PDA differs from the gold standard for more than 16Hz, then we can define:

- Gross Pitch Error:

$$GPE = E_{f0}/N_{voi}$$

- Voiced Detection Error:

$$VDE = (E_{voi \to unv} + E_{unv \to voi})/N_{frame}$$

where N_{voi} is the number of voiced frames in the gold standard and N_{frame} is the number of frames in the utterance. These indicators, taken individually or in pairs, have been used in a large number of works to evaluate the performance of PDAs. The two indicators, however, measure very different errors; it is possible to measure the performance using only one indicator, usually GPE, but it evaluates only part of the problem and hardly provide a faithful picture of PDA behaviour. On the other hand, considering both measures leads to a difficult comparison of the results.

In order to find a remedy to these problems, (Lee and Ellis 2012) suggested slightly different metrics, which allow the definition of a single indicator:

- Voiced Error:

$$VE = (E_{f0} + E_{voi \to unv})/N_{voi}$$

- Unvoiced Error:

$$UE = E_{unv \to voi}/N_{unv}$$

- Pitch Tracking Error:

$$PTE = (VE + UE)/2$$

where N_{unv} is the number of unvoiced frames contained in the gold standard. However, trying to interpret the results obtained by a PDA in light of the PTE measurement is rather complex: it is not immediate to identify from the obtained results the most relevant source of errors.

In light of what has been said previously, it seems appropriate to introduce a new measure of performance that is able to easily capture the performance of a PDA in a single, clear indicator that considers all types of possible errors to be equally relevant. So, following (Tamburini 2013), we adopted the Pitch Error Rate as performance metric, defined as:

$$PER = (E_{f0} + E_{voi \to unv} + E_{unv \to voi})/N_{frame}$$

This measure sum all the types of possible errors without privileging or reducing the contribution of any component and allowing a simpler interpretation of the obtained outcomes.

5. Results

5.1 Preliminary Evaluation

We repeated the same experiments as in (Tamburini 2013) with the Python implementations of the chosen algorithms in order to check the employed codes and to derive common baselines.

Obviously a few small differences in performances will be encountered. Table 1 shows the performance values obtained by the three algorithms compared to all the measures considered for both the gold standards used in the evaluation. We consider these results as baseline performance.

Table 1
The experiments in Tamburini (2013) reproduced using the considered PDA python implementations.

Keele Pitch Database						
PDA	PER	GPE	VDE	PTE	VE	UE
pYAAPT	0.14056	0.05517	0.09777	0.09433	**0.1132**	0.07539
RAPT	**0.12596**	0.04917	**0.08806**	**0.08498**	0.11966	**0.05031**
SWIPE'	0.14236	**0.03556**	0.11474	0.09623	0.12867	0.0638
FDA Corpus						
PDA	PER	GPE	VDE	PTE	VE	UE
pYAAPT	0.11912	0.05381	0.08889	0.08689	0.11016	0.06361
RAPT	**0.09533**	0.03591	**0.07554**	**0.07159**	**0.09637**	**0.0468**
SWIPE'	0.10594	**0.02543**	0.09208	0.07863	0.10652	0.05074

The performances obtained for the FDA corpus are generally better; maybe the algorithms suffer the length of the speech files. As we pointed above, in fact, KPD is a larger corpus with definitely bigger files even if we splitted each of them into four slices. Another important consideration that has to be made, regards the corrupted data in the KPD: removing them from the sequences probably got worse the final evaluation, affecting the total length of the sequences themselves. Furthermore, it has to be kept in mind that we used Python implementations of these algorithms that, as we pointed out some times earlier, are originally available as MATLAB functions. We do not have the proof that this implementation difference affects the results, but more work about checking this issue should be done in the future. Leaving aside these considerations, let us focus on the performances. It can be observed easily that RAPT reaches the best achievements both on KPD and FDA corpus. In evaluating the results obtained, it seems appropriate to study more accurately the types of errors that the three algorithms exhibited in the automatic detection of F0; Table 2 focuses on the total Pitch Error Rate and how this is distributed with respect to the three types of errors that make up its definition, namely E_{f0}, $E_{voi \to unv}$, $E_{unv \to voi}$.

Table 2 shows quite different behaviours among the three pitch detection algorithms: the errors committed seem to be distributed among the different types of error in an uneven way and with different configurations between the PDAs. It could therefore be useful to consider the possibility of combining the contributions of the different algorithms as an attempt to improve their performances. One possibility to do this was to consider, as an estimate of the pitch value in a certain frame, the median of the values calculated by an odd number of different algorithms (in this specific case study, three different PDAs) as it has been done by (Tamburini 2013).

Table 2
Error analysis on the experiments in Tamburini (2013) reproduced using the considered PDA python implementation. We added a further algorithm 'Median', proposed in the cited study, that, for each frame, keeps the median value among the three F0 values proposed by the considered PDAs.

Keele Pitch Database				
PDA	PER	E_{f0}	$E_{voi \to unv}$	$E_{unv \to voi}$
pYAAPT	0.14056	0.04278	0.04411	0.05366
RAPT	0.12596	0.03789	0.05252	**0.03554**
SWIPE'	0.14236	0.02762	0.06985	0.04488
Median	**0.08814**	**0.02656**	**0.03359**	0.03564
FDA Corpus				
PDA	PER	E_{f0}	$E_{voi \to unv}$	$E_{unv \to voi}$
pYAAPT	0.11912	0.03023	**0.03399**	0.0549
RAPT	0.09533	0.01978	0.03438	0.04116
SWIPE'	0.10594	**0.01385**	0.04773	0.04434
Median	**0.10182**	0.02537	0.03686	**0.03917**

From Table 2 it emerges quite clearly how the combination of different algorithms with the median method makes better results. In particular, it is worth underlying how much the E_{f0} error decreases, especially in the experiments involving KPD.

This section presented an objective evaluation of three algorithms for the automatic extraction of the fundamental frequency value in the spoken language, using a large set of different metrics. It will be useful as a baseline for comparing the performances of the proposed neural PDA smoother.

5.2 Neural PDA Evaluation

In order to carry out an objective evaluation of our pitch smoother, we decided to put our attention on one of the metrics employed for the evaluation of the three Pitch Detection Algorithms, namely the Pitch Error Rate (PER). In fact, as we pointed out earlier, this measure is able to easily capture the performance of a PDA in a single, clear indicator that considers all types of possible errors to be equally relevant.

After the influential paper from (Reimers and Gurevych 2017) it is clear to the community that reporting a single score for each DNN training session could be heavily affected by the system initialisation point and we should instead report the mean and standard deviation of various runs with the same setting in order to get a more accurate picture of the real systems performances and make more reliable comparisons between them.

The PER metric was computed for each epoch during the training phase for all subsets in order to determine the stopping epoch when we get the minimum PER on the validation set. We performed 10 runs for each experiment computing means, standard deviations and significance tests.

We also tested our pitch smoother on the mixed configurations of the datasets employed, adopting the same procedures.

Table 3 shows all the obtained results. The proposed system always exhibits the best results in any experiment with relevant performance gains with respect to the PDAs base outputs. All the differences resulted highly significant when applying a t-test. Given the very small standard

deviation in all the experiments we can conclude that, in this case, the initialisation point did not affect the neural network performances too much.

Table 3
PER mean (μ) and standard deviation (σ) obtained by the proposed pitch profile smoother. One sample t-test significance test returns p$\ll$0.001 for all experiments. N.B.: Even if the number of experiments is small (10), the power analysis of the t-tests is always equal to 1.0 showing maximum t-test reliability. The assumption of normality has been tested, with the Shapiro-Wilk test, before computing the t-test.

Keele Pitch Database			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.14056	**0.07958**	0.00271
RAPT	0.12596	**0.08481**	0.00376
SWIPE'	0.14236	**0.10065**	0.00292
FDA Corpus			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.11912	**0.06731**	0.00421
RAPT	0.09533	**0.06752**	0.00232
SWIPE'	0.10594	**0.07769**	0.00212
Mixed Keele+FDA Corpus			
PDA	PDA PER	Smoother PER μ	Smoother PER σ
pYAAPT	0.06951	**0.06302**	0.00246
RAPT	0.09859	**0.07256**	0.00297
SWIPE'	0.08758	**0.08151**	0.00144

Referring to the performance outcomes of the Pitch Detection Algorithms we provided in Table 3, it can be easily noted a general, great improvement. For both the configurations we employed, pYAAPT shows the best performances; the category in which we observe the bigger error in each of our combinations is $E_{voi \to unv}$, the number of frames erroneously classified between voiced and unvoiced; this means that our smoother has a major struggle in correctly identifying the boundaries between voiced and unvoiced regions. Despite this, our pitch smoother behaves rather well in correcting all halving and doubling errors, which are collected in E_{f0}, the indicator that measures the error of estimation of the F0 values on frames considered voiced.

We performed a one sample t-test significance test that returned $p \ll 0.001$ for all experiments and, even if the number of experiments is small (10), the power analysis of the t-tests was always equal to 1.0, showing maximum t-test reliability.

6. Conclusions

This paper presented a new pitch smoother based on recurrent neural networks that obtained excellent results when evaluated using two standard benchmarks for English. The results showed that our smoother is able to efficiently learn how to smooth a pitch profile produced by widely used PDAs removing halving and doubling errors from the profile. The proposed Neural

Smoother will not increase the total processing time for each utterance as, once properly trained, is able to process and correct a single intonation profile very quickly.

Future works could regard the intermixing of various corpora in different languages in order to test the possibility of deriving a pitch smoother able to properly work without caring about language and, possibly, specific corpora and language registers. In principle we can imagine that it would be possible to train a neural pitch smoother like the one presented in this paper cross-linguistically to correct the pitch detection errors and apply it to smooth the PDAs profiles obtained on different languages and registers. This is a pure speculation and we definitively have to perform new experiments in order to verify this idea. The main problem in performing such experiments is the availability of speech corpora provided with the laryngograph profiles. We need definitely a good sample of, at least, different languages to perform these experiments and, at the time of writing, we have only few corpora of this kind.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Asgari, Meysam and Izhak Shafran. 2013. Improving the accuracy and the robustness of harmonic model for pitch estimation. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2013*, pages 1936–1940, Lyon, France, August.

Babacan, Onur, Thomas Drugman, Nicolas D'Alessandro, Nathalie Henrich, and Thierry Dutoit. 2013. A comparative study of pitch extraction algorithms on a large variety of singing sounds. In *IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2013*, pages 7815–7819, Vancouver, Canada, May.

Bagshaw, Paul C. 1994. *Automatic prosodic analysis for computer-aided pronunciation teaching*. Ph.D. thesis, University of Edimburgh.

Bagshaw, Paul C., Steven M. Hiller, and Mervyn A. Jack. 1993. Enhanced pitch tracking and the processing of f0 contours for computer aided intonation teaching. In *Proceedings of Eurospeech '93*, pages 1003–1006, Berlin, September.

Bartosek, Jan. 2010. Pitch detection algorithm evaluation framework. In *Proceedings of 20th Czech-German Workshop on Speech Processing*, pages 118–123. Prague, Czech Republic, September.

Brøndsted, Tom. 1997. Intonation contours 'distorted' by tone patterns of stress groups and word accent. In *Intonation: Theory, Models and Applications: Proceedings of an ESCA Workshop*, pages 55–58, Athens, Greece, September.

Byrd, Dani and Toben H.Mintz. 2010. *Discovering Speech, Words, and Mind*. Wiley-Blackwell.

Camacho, Arturo. 2007. *SWIPE: A sawtooth waveform inspired pitch estimator for speech and music*. Ph.D. thesis, University of Florida.

Chu, Wei and Abeer Alwan. 2009. Reducing f0 frame error of f0 tracking algorithms under noisy conditions with an unvoiced/voiced classification frontend. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2009*, pages 3969–3972, Taipei, Taiwan, April.

Chu, Wei and Abeer Alwan. 2012. Safe: A statistical approach to f0 estimation under clean and noisy conditions. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):933–944.

de Cheveigné, Alain and Hideki Kawahara. 2002. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930.

Gawlik, Mateusz and Wieslaw Wszolek. 2018. Modern pitch detection methods in singing voices analyzes. In *Proceedings of Euronoise 2018*, pages 247–253, Crete, May.

Gonzalez, Sira and Mike Brookes. 2014. PEFAC-A pitch estimation algorithm robust to high levels of noise. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(2):518–530.

Han, Kun and DeLiang Wang. 2014. Neural network based pitch tracking in very noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 22(12):2158–2168.

Hochreiter, Sepp and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huang, Feng and Tan Lee. 2012. Robust pitch estimation using l1-regularized maximum likelihood estimation. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 378–381, Portland (OR), September.

Hui, Lu, Lu Hui Ting, Swee Lan See, and Paul Y. Chan. 2015. Use of electroglottograph (EGG) to find a relationship between pitch, emotion and personality. *Procedia Manufacturing*, 3:1926–1931.

Jang, Seung-Jin, Seong-Hee Choi, Hyo-Min Kim, Hong-Shik Choi, and Young-Ro Yoon. 2007. Evaluation of performance of several established pitch detection algorithms in pathological voices. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society - EMBC 2007*, pages 620–623, Lyon, France, August.

Jin, Zhaozhang and DeLiang Wang. 2011. Hmm-based multipitch tracking for noisy and reverberant speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1091–1102.

Jlassi, Wided, Aicha Bouzid, and Noureddine Ellouze. 2016. A new method for pitch smoothing. In *2nd International Conference on Advanced Technologies for Signal and Image Processing*, pages 657–661, Monastir, Tunisia, March.

Jouvet, Denis and Yves Laprie. 2017. Performance analysis of several pitch detection algorithms on simulated and real noisy speech data. In *25th European Signal Processing Conference - EUSIPCO 2017*, pages 1614–1618, Kos, Greece, September.

Kellman, Michael R. and Nelson Morgan. 2017. Robust multi-pitch tracking: a trained classifier based approach. Technical report, ICSI Tchnical Report, Berkeley, CA.

Kingma, Diederik P. and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations - ICLR 2015*, San Diego, CA, May.

Kotnik, Bojan, Harald Höge, and Zdravko Kacic. 2006. Evaluation of pitch detection algorithms in adverse conditions. In *Proceedings of Speech Prosody 2006*, PS2-8-83, Dresden, May.

Lee, Byung Suk and Daniel P. W. Ellis. 2012. Noise robust pitch tracking by subband autocorrelation classification. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 707–710, Portland (OR), September.

Luengo, Iker, Ibon Saratxaga, Eva Navas, Inmaculada Hernaez, Jon Sanchez, and Inaki Sainz. 2007. Evaluation of pitch detection algorithm under real conditions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2007*, pages 1057–1060, Honolulu, Hawaii, April.

Murray, Kathleen. 2001. A study of automatic pitch tracker doubling/halving errors. In *Proceedings of 2nd SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark, September.

Plante, Fabrice, Georg F. Meyer, and William A. Ainsworth. 1995. A pitch extraction reference database. In *Proceedings of Eurospeech '95*, pages 837–840, Madrid, September.

Rabiner, Lawrence R., Michael J. Cheng, Aaron E. Rosenberg, and Carol A. McGonegal. 1976. A comparative performance study of several pitch detection algorithms. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 24(5):399–418.

Reimers, Nils and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of Conference on Empirical Methods in Natural Language Processing - EMNLP 2017*, pages 338–348, Copenhagen, Denmark, September.

Shriberg, Elizabeth E. 1999. Phonetic consequences of speech disfluency. In *Proceeedings of the International Congress of Phonetic Sciences - ICPhS '99*, pages 619–62, San Francisco, August.

So, YongJin, Jia Jia, and LianHong Cai. 2012. Analysis and improvement of auto-correlation pitch extraction algorithm based on candidate set. In Q. Zhihong, C. Lei, S. Weilian, W. Tingkai, and Y. Huamin, editors, *Recent Advances in Computer Science and Information Engineering: Volume 5*. Springer, Heidelberg/Berlin, pages 697–702.

Stylianou, Yannis. 1996. *Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification*. Ph.D. thesis, Ecole Nationale Superieure des Telecommunications.

Sukhostat, Lyudmila and Yadigar Imamverdiyev. 2015. A comparative analysis of pitch detection methods under the influence of different noise conditions. *Journal of Voice*, 29(4):410–417.

Talkin, David. 1995. A robust algorithm for pitch tracking (RAPT). In W. B. Kleijn and K. K. Paliwal, editors, *Speech Coding and Synthesis*. Elsevier, New York, pages 495–518.

Tamburini, Fabio. 2013. Una valutazione oggettiva dei metodi più diffusi per l'estrazione automatica della frequenza fondamentale. In *Atti dell IX Convegno Nazionale dell'Associazione Italiana di Scienze della Voce - AISV 2013*, pages 427–434, Roma, January.

Titze, Ingo R. 1994. *Principles of Voice Production*. Prentice Hall.

Veprek, Peter and Michael S. Scordilis. 2002. Analysis, enhancement and evaluation of five pitch determination techniques. *Speech Communication*, 37(3-4):249–270.

Wang, Dongmei and Philipos C. Loizou. 2012. Pitch estimation based on long frame harmonic model and short frame average correlation coefficient. In *Proceedings of 13th Annual Conference of the International Speech Communication Association - Interspeech 2012*, pages 923–926, Portland, OR, September.

Wu, Mingyang, DeLiang Wang, and Guy J. Brown. 2003. A multipitch tracking algorithm for noisy speech. *IEEE Transactions on Audio, Speech, and Language Processing*, 11(3):229–241.

Zahorian, Stephen A. and Hongbing Hu. 2008. A spectral/temporal method for robust fundamental frequency tracking. *Journal of the Acoustical Society of America*, 123(6):4559–4571.

Zhao, Xufang, Douglas O'Shaughnessy, and Nguyen Minh-Quang. 2007. A processing method for pitch smoothing based on autocorrelation and cepstral f0 detection approaches. In *Proceedings of the International Symposium on Signals, Systems and Electronics*, pages 59–62, Montreal, Canada, August.