Direct F0 Estimation with Neural-Network-based Regression

Citation for published version:

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proc. Interspeech 2019

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Direct F0 Estimation with Neural-Network-based Regression

Shuzhuang Xu¹, Hiroshi Shimodaira²

¹School of Informatics, University of Edinburgh
²Centre for Speech Technology Research, University of Edinburgh

sz.xu@outlook.com, h.shimodaira@ed.ac.uk

Abstract

Pitch tracking, or the continuous extraction of fundamental frequency from speech waveforms, is of vital importance to many applications in speech analysis and synthesis. Many existing trackers, including conventional ones such as Praat, RAPT and YIN, and newly proposed neural-network-based ones such as DNN-CLS, CREPE and RNN-REG, have conducted an extensive investigation into speech pitch tracking. This work developed a different end-to-end regression model based on neural networks, where a voice detector and a newly proposed value estimator work jointly to highlight the trajectory of fundamental frequency. Experiments on the PTDB-TUG corpus showed that the system surpasses canonical neural networks in terms of gross error rate. It further outperformed conventional track-
to a balancing point between classification and regression, incapable of maximising its power on each sub-task. Some models transform the whole task into a pure classification, therefore avoid this conflict: however, it still brings new constraints as described in Introduction.

Based on the consideration, we supposed the shared hidden representation may prevent from better performance and proposed to use a pair of dedicated F0 detector and estimator. For each frame input, the detector determines the F0 existence: if voice F0 is present, the estimator then evaluates the F0 value; otherwise, a zero is produced to be compatible with conventional systems. Under this framework, the detector and estimator are only involved in one sub-task respectively and can be trained independently.

2.1. Voice detection

The voice detection is formalised as a binary classification. The probability of a frame being either voice or unvoice frame is represented in two states respectively. Therefore, the target value is 1 or 0 instead of F0. Accordingly, activation function e.g. sigmoid is applicable for the output layer and loss e.g. binary cross-entropy loss is compatible for training.

We implemented fully-connected deep feedforward network (FDNN) as well as its improvement by introducing dropout [12] and batch normalisation [13]. Dropout is a simple but effective technique that partially masks layer units as zeros in training time, attenuating the co-dependency between units and hence constraining overfit. Batch normalisation is a regularisation technique that re-scales tensors in a mini-batch by learnable parameters to centralise values and stabilise variance, reducing the network difficulty to learn new distributions of incoming tensors. Apart from this, we explored the possibility by the more recently innovated residual network on this classification. Residual network introduces the connection between non-adjacent layers and changes the learning objective of the vaulted layer to the difference between source and target. A pair of popular implementations including the earlier model [13] and the later model [14] are implemented.

In addition, simple pre-processing is used. Since adjacent frames may include supportive information to determine F0 existence, the length l of the input frame x is enlarged to p times by symmetrically enclosing sampling points from both sides. Apart from this, since the loudness varies among recordings, each whole recording X is normalised to \( X' = X - \frac{\sum |X|}{M(|X|)} \) where \( M(\cdot) \) computes the absolute average value. Note that the pre-processing above is applied for the voice detector only.

2.2. F0 estimator

The F0 estimation is formalised as a numerical regression. Though the networks for voice detection should be applicable as an F0 estimator by altering the output layer to ReLU activation [15], we conduct a further investigation to seek better improvement. Inspired by “jump wire” techniques such as highway and residual networks that shorten the distance between shallow layers and deeper layers, as well as how an exponential series represents a natural number \( n = \sum_{i=0}^{\infty} \left(c_i \times 2^i\right) \), where \( c_i \in \{0, 1\} \), we consider every F0 value \( \hat{f} \) instead of being derived from a single output layer, can be assembled from multiple output nodes:

\[
\hat{f} = \sum_{i=1}^{n} k_i \cdot o_i + b
\]

where \( o_1, o_2 \cdots o_n \in [0, 1] \) are values from output nodes with sigmoid activation; \( k_i, b \) and \( n \) are pre-defined constants such that the equation’s lower and upper bound \([b, b + \sum_{i=1}^{n} k_i]\) covers the practical F0 range \([f_{\text{min}}, f_{\text{max}}]\). We then proposed a network, value decoder (VD), as illustrated in Figure 1. where the outputs \( o_1, o_2 \cdots o_n \) obtained in a forward propagation are given by:

\[
o_i = O(H_{\text{1}}(\cdots H_{\text{2}}(H_{\text{1}}(F(x)) \cdots )))
\]

where: \( x \) is the input frame from a whole recording \( X; F(\cdot) \) is an arbitrary combination of layers including the input layer; \( H_{\text{1}}(\cdot), H_{\text{2}}(\cdot) \cdots H_{\text{1}}(\cdot) \) are hidden layers of either regular ones e.g. fully-connected, or RNN cells e.g. LSTM [16] and GRU [17]; and \( O(\cdot) \) is the output layer with shared parameters and sigmoid activation to derive \( o_1, o_2 \cdots o_n \).

Figure 1: The diagram of the proposed value decoder

This model makes the final F0 no longer inferred from a single output layer but a computation from layers in different depths i.e. a distributed representation. It can be trained by a regular numerical loss such as \( L_1 \) and mean squared error (MSE) loss.

3. Experiment

The system performance is evaluated in terms of accuracy on voice detection and value estimation. Comparison is made vertically and horizontally: vertically, the candidate models for detector and estimator are compared; horizontally, the system assembled with the best detector and estimator is compared with external models.

3.1. Pitch tracking corpus

The latest pitch tracking corpus from the Graz University of Technology (PTDB-TUG) [18] is adopted for experiments, which surpasses previous corpora in terms of quality and abundance. This database consists of 236 sentences selected from the TIMIT corpus based on phonetic richness. Each sentence is spoken by 10 women and 10 men, giving a total of 4720 pieces of recordings. The reference F0 values are derived from the coupled laryngograph, labelled for every frame of 32 milliseconds (adjacent frame is overlapped by 22 milliseconds). The total number of training pairs is 3420k and 807k of which are of voice. F0 in the database covers a wide range from about 50Hz to 380Hz, with mean, median and standard deviation at around 151Hz, 148Hz and 52Hz respectively. The only pre-processing on the database is a down-sampling to 12kHz instead of using the original 48kHz sampling rate. For dataset division, 80% of the total are formed into a 4-fold cross validation set for training and validation, and 20% is held-out for test. In addition, the robustness is investigated under a simulated noise condition by mixing standard noise from the NOISEX-92 corpus [19] onto...
the clean waveform. 5 different noises, babble, factory1, destroyerops, leopard and white, selected based on sound characteristic, are applied at 3 different levels of signal-to-noise ratio (SNR): -10dB, 0dB and 10dB.

3.2. Experiment configuration

For the voice detector, it is trained on all training pairs including both voice and unvoice frames. For the F0 estimator, it is trained solely on voice pairs as the hypothesis suggests unvoice frames are not suitable for estimator. L1 loss is adopted in order to better constrain the overall inaccuracy, instead of MSE loss which will better constrain pairs with larger errors. The window enlarging ratio $q$ is 3 for detector and 1 for estimator (i.e. not changed).

In the vertical benchmark, experiment is first conducted to find the optimal hyperparameters $k$, $b$ and $n$ for the proposed value decoder, where $b = 0$ and $k, n = \{100, 4; 133, 3; 200, 2; 400, 1\}$ that all match a range between 0Hz and 400Hz are tested. Then, the value decoder with the best hyperparameters is compared with other neural networks with key specifications listed in Table 1. As a note, the model FDNN-3 is trained to work on both voice and unvoice frames. Therefore, it has only one unit for the output layer, where the output is the F0 value for voice frames and 0 for unvoice frames by ReLU (any output lower than 30Hz is also regarded as 0Hz). The purpose of this particular model is to validate the hypothesis that using separate models will improve the performance.

For the horizontal benchmark, the assembled model with the best detector and estimator is compared with conventional F0 trackers including Praat, RAPT and YIN under clean condition as well as the representative classifier DNN-CLS [9] under simulated noisy condition. The DNN-CLS here is modified to have the same structure as FDNN-2 (the width of the input layer is 1024) and a frequency state for every integer rounded from reference F0 plus one state for unvoice). Anything else is implemented as the original configuration as best as possible, except that the input waveform is adjusted to use the original width and interval from the PTDB-TUG corpus.

3.3. Evaluation metrics

Pitch tracking performance is evaluated on standard metrics:

- **Voice decision error (VDE):** This includes all frames which are incorrectly classified, either voice frames classified as unvoice or unvoice as voice.
- **Gross pitch error (GPE):** This includes all voice frames with a relative error larger than 20% of its reference F0. A narrow definition is used here that gross pitch error excludes any frames that already trigger VDE.
- **F0 frame error (FFE):** This includes all frames that trigger either VDE or GPE, which can be used as a combined measure of the system performance. Note that the rate of FFE equals to the sum of VDE and GPE.
- **Fine pitch error (FPE):** This includes all remaining voice frames that triggers neither VDE nor GPE. The average $\mu_{FPE}$ and standard deviation $\sigma_{FPE}$ of the absolute difference between the derived and referencing F0 describe the level of value accuracy on F0 estimation.

Table 1: Network specification for vertical comparison

<table>
<thead>
<tr>
<th>Models</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDNN-1</td>
<td>FDNN with 6 hidden layers of 1024, 896, 768, 640, 448 and 384 units respectively using ReLU activation</td>
</tr>
<tr>
<td>FDNN-2</td>
<td>Same as FDNN-1 but each hidden layer has 25% dropout and batch normalisation</td>
</tr>
<tr>
<td>FDNN-3</td>
<td>Same as FDNN-1 but used for voice detection and F0 estimation in a single model</td>
</tr>
<tr>
<td>ResNet-1</td>
<td>3 residual cell [13] of width 1024</td>
</tr>
<tr>
<td>ResNet-2</td>
<td>3 residual cell [14] of width 1024</td>
</tr>
<tr>
<td>VD-FNN</td>
<td>Value decoder using FDNN (width 1024)</td>
</tr>
<tr>
<td>VD-RNN</td>
<td>Value decoder using LSTM (width 1024)</td>
</tr>
</tbody>
</table>

Table 2: Value decoders with different hyperparameters

<table>
<thead>
<tr>
<th>(b=0)</th>
<th>n=1</th>
<th>n=2</th>
<th>n=3</th>
<th>n=4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$k=400$</td>
<td>$k=200$</td>
<td>$k=133$</td>
<td>$k=100$</td>
</tr>
<tr>
<td>GPE</td>
<td>2.81</td>
<td>2.33</td>
<td>2.21</td>
<td>2.14</td>
</tr>
</tbody>
</table>

Using the best configuration above, two variants of value decoder are further implemented in vertical comparison: using FDNN and LSTM as the cell respectively. The result of the vertical benchmark for all candidates as voice detector or F0 estimator is presented in Figure 3. The comparison between FDNN-2 and FDNN-3 shows that all performance metrics degrades in the latter, indicating an accurate regression from raw waveform to numerical F0 might not be achievable by the shared model, which proves the feasibility by using dedicated models for F0 detection and estimation. In other words, our hypothesis, that the shared hidden representation for F0 detection and estimation is an obstacle for better performance, holds in the context of direct numerical regression.

Table 3: Vertical comparison across candidate models

<table>
<thead>
<tr>
<th>Models</th>
<th>Detector Estimator</th>
<th>VDE</th>
<th>GPE</th>
<th>$\mu_{FPE}$</th>
<th>$\sigma_{FPE}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDNN-1</td>
<td></td>
<td>3.85</td>
<td>2.73</td>
<td>2.59</td>
<td>3.19</td>
</tr>
<tr>
<td>FDNN-2</td>
<td></td>
<td>3.40</td>
<td>2.51</td>
<td>2.15</td>
<td>3.11</td>
</tr>
<tr>
<td>FDNN-3</td>
<td></td>
<td>3.51</td>
<td>3.10</td>
<td>4.57</td>
<td>3.90</td>
</tr>
<tr>
<td>ResNet-1</td>
<td></td>
<td>3.82</td>
<td>2.36</td>
<td>1.97</td>
<td>3.10</td>
</tr>
<tr>
<td>ResNet-2</td>
<td></td>
<td>3.74</td>
<td>2.57</td>
<td>2.40</td>
<td>3.47</td>
</tr>
<tr>
<td>VD-FNN</td>
<td></td>
<td>-</td>
<td>2.14</td>
<td>1.47</td>
<td>2.66</td>
</tr>
<tr>
<td>VD-RNN</td>
<td></td>
<td>-</td>
<td>2.44</td>
<td>1.42</td>
<td>2.63</td>
</tr>
</tbody>
</table>

Among these dedicated models for detection and estimation, it is obvious that the FDNN-1 presents the overall worst performance, while other models all present some improvement. For F0 detection, it shows that both residual network structures have very little help for better determination on F0 existence with ignorable difference. On the contrary, dropout and batch normalisation efficiently improves voice detection by 0.45% dropping in VDE. For F0 estimation, it shows that the residual network proposed by [13] achieves all metrics better than DNN with dropout and batch normalisation, while the alternative residual structure proposed by [14] underperforms the former. However, all these models mentioned above does not
it also proves that the shared hidden representation degrades the model is identical to the voice detector in the proposed system, since the hidden structure of the modified DNN-CLS classification model in both model capability and noise robustness. This result indicates that the dedicated voice detection model in the proposed model under clean condition (Praat).

Figure 2: The comparison between conventional models and the proposed model under clean condition

Figure 3 demonstrates the result of the modified DNN-CLS model as well as the proposed model under simulated noisy condition. With respect to voice decision accuracy, it shows that the proposed system outperforms other contrastive ones. The improvement is accumulated from better VDE which decreases by around 25% compared with RAPT, as well as better GPE which is almost 1/3 of that of YIN, to an overall FFE 30% better than other best performed model (Praat).

Last but not least, the mechanism of the proposed value decoder deserves a discussion. Some samples from a trained value decoder with \( k = 100, b = 0 \) and \( n = 4 \) is presented in Table 4. As expected, the inferred \( F_0 \) is not derived from a single output layer, instead, from distributed output nodes. There are 2 functional parts in this structure: “subtractor” and “estimator”. The hidden layers \( H_1(\cdot), H_2(\cdot), \cdots, H_n(\cdot) \) are trained to be subtractors. Through each subtractor, an abstract hidden representation for \( k \)Hz is subtracted from the incoming tensor. The shared output layer \( O(\cdot) \) is trained to be a numerical estimator. If the hidden vectors contain a frequency higher than \( k \)Hz or “negative” frequency, the \( \text{sigmoid} \) activation can prevent the value from out-of-control. This structure has some interesting similarity with known models: it introduces shortcut connections from shallow layers to deep layers, as Highway network and residual network; and it visually looks like a time-flattened decoder in a canonical sequence-to-sequence model, while employing no loops from the output back to the input. By employing this design, shallow layers in this structure also have output layer closely connected, allowing back-propagation functions more effectively. The difficulty in estimating an accurate value in a time is decomposed into multiple output nodes and more instructive hints can be backpropagated to layers. They are considered as the key reasons of its success.

4. CONCLUSION

Our work conducts a successful investigation on how to directly estimate speech \( F_0 \) from raw waveform with neural networks. It concludes that using dedicated models for voice detection and value estimation is a positive strategy to improve the overall performance. For voice detection, DNN with dropout and batch normalisation is shown to be very efficient. For \( F_0 \) estimation, we proposed a good-performing value decoder structure, which outperforms both DNN and residual networks on all performance metrics. Constructed with the best components, the final system significantly outperforms traditional models in terms of both voice decision and pitch accuracy. Compared with representative neural-network classifier for pitch tracking, the system still preserves a competitive accuracy on voice decision under very noised condition; however, the value accuracy of the value decoder is not as robust as the contrastive model. By the decomposition in the pitch tracking task as well as the proposed value decoder, we demonstrated the effectiveness of decomposing a complex task into simpler tasks, which is also applicable to other tasks. The feasibility of the proposed value decoder on other regression task is an open area of research.

Table 4: Value decoder behaviours

<table>
<thead>
<tr>
<th>( F_0 ) (Ref.)</th>
<th>( \alpha_1 )</th>
<th>( \alpha_2 )</th>
<th>( \alpha_3 )</th>
<th>( \alpha_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>66</td>
<td>0.65</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>151</td>
<td>0.97</td>
<td>0.53</td>
<td>0.01</td>
<td>0.00</td>
</tr>
<tr>
<td>234</td>
<td>1.00</td>
<td>0.96</td>
<td>0.42</td>
<td>0.00</td>
</tr>
<tr>
<td>309</td>
<td>1.00</td>
<td>0.99</td>
<td>0.75</td>
<td>0.32</td>
</tr>
</tbody>
</table>
5. References


