MIREX 2019: MULTIPLE F_0 ESTIMATION USING ECHO STATE NETWORKS

Peter Steiner
Technische Universität Dresden
Germany
peter.steiner@tu-dresden.de

Azarakhsh Jalalvand
Ghent University
Belgium

Peter Birkholz
Technische Universität Dresden
Germany

ABSTRACT

This extended abstract describes the SJB1 $-$ 4 submission for the MIREX multiple-f_0 estimation challenge. The model utilized here is an echo state network (ESN), a variation of recurrent neural networks (RNNs). The fundamental difference to typical RNN architectures is that the input weights and the recurrent connection weights are initialized by random values, and only the output weights are trained using linear regression. Input to the ESN were spectral feature vectors, calculated with a bank of logarithmically-spaced frequency filters, normalized over time. The normalized feature vectors were fed into the ESN, which computed an output vector each MIDI pitch and an additional silence output. This was a representation for the pitch and silence states for each frame. All pitch states above a global threshold represented active pitches and vice versa.

1. INTRODUCTION

The fundamental frequency f_0 is the smallest noticable frequency in a quasiperiodic signal, and related to the perceived pitch p. Detecting the f_0 in a speech signal is nowadays a well-investigated task. Algorithms implemented in software packages, such as Praat [1] or RAPT [8], as well as YIN-based techniques [2, 5, 7] are able to extract the f_0 of speech or monophonic musical instruments without a high computational complexity and without the need for training data.

In music, however, multiple sources can be active at the same time, and each source has its own f_0. Thus, the goal of algorithms for *multiple-f_0* extraction is to extract all f_0 values present at any time. Main challenges in developing such algorithms are unknown polyphony (number of active notes or sources during the same time), different and unknown instruments, and fast note transitions. A common way of simplifying this task is to map the f_0 values to a reduced set of center frequencies of the notes in the western music. In this context, multiple-f_0 estimation can also be considered as a multilabel classification, where each note is a separate class.

Echo State Networks (ESNs) by Herbert Jaeger [3] are a special kind of Recurrent Neural Networks (RNNs) and have achieved comparable results to CNNs in several recognition tasks, such as speech and image recognition [4, 10]. However, they are fairly new in the context of Music Information Retrieval (MIR). Encouraged by the positive performance in various research areas, this work explored the potential of ESNs for the challenging task of multiple-f_0 estimation. ESNs have some beneficial properties for this task:

- They are suitable for processing temporal information due to recurrent connections, and the training procedure is much easier than concurrent approaches due to less free parameters.
- They are quite robust against noise and unseen conditions.
- Since the input dimension has minimum (almost non) impact on the complexity of the model, they are interesting candidates for processing high dimensional data.

2. MULTIPLE F_0 ESTIMATION WITH ECHO STATE NETWORKS

The main outline of our proposed ESN-based model for multiple-f_0 estimation is depicted in Figure 1. All Figures, explanations and Tables are taken from a paper that will be published shortly. [6]

2.1 Echo State Network (ESN)

The main outline of an ESN is depicted in the center of Figure 1. It consists of the input weights $\mathbf{W}^{\mathrm{in}}$, the reservoir weights $\mathbf{W}^{\mathrm{res}}$ and the output weights $\mathbf{W}^{\mathrm{out}}$.

The input weight matrix $\mathbf{W}^{\mathrm{in}}$ has the dimension of $N^{\mathrm{in}} \times N^{\mathrm{res}}$ where $N^{\mathrm{in}} = 512$ and N^{res} are the size of the input feature vector and the size of the reservoir, respectively. All values in this matrix were initialized from a standard normal distribution. Next, each node of the reservoir was only connected to $K^{\mathrm{in}} = 10$ randomly selected input entries. The other connections were set to zero, leading to a very sparse matrix $\mathbf{W}^{\mathrm{in}}$. The input weight matrix was then scaled using the input scaling factor α_{U}, which was a hyper-parameter to be tuned.

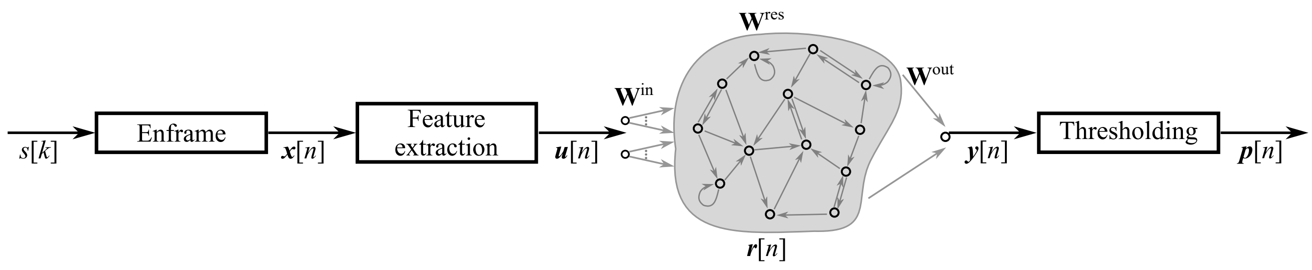

Figure 1: Outline of the ESN-based proposed model: The input signal $s[k]$ with the sample index k was divided into overlapping frames, from which normalized feature vectors $\mathbf{u}[n]$ were extracted and fed into the reservoir using the input weight matrix $\mathbf{W}^{\text{in}}$. The reservoir consists of unordered and via the reservoir matrix $\mathbf{W}^{\text{res}}$ sparsely connected neurons. The output vector $\mathbf{y}[n]$ with $N^{\text{out}} = 129$ dimensions is a linear combination of the reservoir states $\mathbf{r}[n]$ and the output weight matrix $\mathbf{W}^{\text{out}}$, which was trained using linear regression. Each component of the output vector $\mathbf{y}[n]$ corresponded to one musical note, and one additional component to silence. Using a threshold, the output was converted to a binary sequence of notes $\mathbf{p}[n]$.

The reservoir weight matrix $\mathbf{W}^{\text{res}}$ is a square matrix of the size $N^{\text{res}} \times N^{\text{res}}$, which was also initialized from a standard normal distribution. Each reservoir node received values from only $K^{\text{rec}} = 10$ randomly selected other nodes. The other connections were set to zero. The reservoir matrix $\mathbf{W}^{\text{res}}$ was normalized by its largest absolute eigenvalue to achieve a spectral radius $\rho = 1.0$, because it was shown in [3] that the echo state property holds as long as $\rho \leq 1.0$. By tuning α_{U} and ρ, it is possible to balance, how strongly the network memorizes past inputs compared to the present input.

If $\mathbf{r}[n]$ represents the reservoir state, the basic equations to describe the ESN can be written in the following way:

$$\mathbf{r}[n] = (1-\lambda)\mathbf{r}[n-1] + \lambda f_{\text{res}}(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n-1]) \tag{1}$$

$$\mathbf{y}[n] = \mathbf{W}^{\text{out}}\mathbf{r}[n] \tag{2}$$

Equation (1) is a leaky integration of the reservoir neurons. Depending on the leakage $\lambda \in [0, 1]$, the reservoir can act as a long-term or a short-term memory. The reservoir activation function $f_{\text{res}}(\cdot)$ controls the non-linearity of the system. Here, the $\tanh$-function was used, because its lower and upper boundaries of ± 1 ensure stable reservoir states.

Equation (2) shows how to compute the N^{out}-dimensional output vector $\mathbf{y}[n]$ from a given reservoir state $\mathbf{r}[n]$, which was expanded by one bias term. The hyper-parameter α_B was used to scale the impact of this bias term. The output is obtained by a linear combination of the reservoir state and the output weight matrix $\mathbf{W}^{\text{out}}$. For training, all reservoir states were collected in the reservoir state collection matrix $\mathbf{R}$, and expanded by one bias term.The desired output vectors $\mathbf{d}[n]$ were collected into the desired output collection matrix $\mathbf{D}$. Afterwards, $\mathbf{W}^{\text{out}}$ was obtained using regularized linear regression (3), i.e. ridge regression to prevent overfitting to the training data. The regularization parameter $\epsilon = 0.0001$ penalized large values in $\mathbf{W}^{\text{out}}$, and $\mathbf{I}$ is the identity matrix.

$$\mathbf{W}^{\text{out}} = \left(\mathbf{R}\mathbf{R}^{\text{T}} + \epsilon\mathbf{I}\right)^{-1}\left(\mathbf{D}\mathbf{R}^{\text{T}}\right) \tag{3}$$

2.2 Bidirectional and stacked reservoirs

In the case of bidirectional reservoirs, the input was first fed through the ESN as described before. Before the linear regression, the inputs were reversed in time, and again fed into the same reservoir. Afterwards, the reservoir states were again reversed in time. The reservoir state collection matrix $\mathbf{R}$ was finally built by combining the states from the forward and backward pass. This doubled the number of free parameters for the linear regression. For example, the number of features for a reservoir with 500 neurons is 500 in the unidirectional and 1000 in the bidirectional case. The final training remained the same as before.

In the case of stacked reservoirs, the layers were trained sequentially using the same desired outputs in every layer. After fixing the hyper-parameters for one layer, the output of that served as the input for the next layer. By stacking reservoirs, some mistakes from one layer can be corrected in the next layer, because it offers additional temporal modeling capacity. This can be done for unidirectional as well as for bidirectional reservoirs.

3. EXPERIMENTAL SETUP

3.1 Dataset

To evaluate the capabilities of Echo State Networks to transcribe music, the recently introduced MusicNet database [9] was used, which is the largest freely available database. It contains in total 330 classical music recordings, using 11 different instruments. All recordings are mono audio files and sampled with $f_s = 44\,100\,\text{Hz}$. In total, there are more than 30 h of music with sample-based annotations for instruments, notes and more.

The MusicNet is by default split into a 320 training and 10 test files. The hyperparameters were tuned solely on the training set, and the test files were just used one time in the end to report measurements.

3.2 Optimization

The algorithm was developed in Python 3, and was based on [4]. Table 1 shows the optimized hyper-parameters for the uni- and bidirectional models with one and two layer architectures.

Hyperparameter	Final values	
	Layer 1	Layer 2
Input scaling α_{U}	0.1	1.9
Spectral radius ρ	0.8	0.1
Bias scaling α_{B}	1.8 (u) 0.85 (b)	1.35
Leakage λ	0.1	0.2
Threshold δ	0.35	

Table 1: Overview over all hyperparameters that were trained with the MusicNet dataset. The final values of every model were fixed for the evaluation. The optimization lead to equal values for unidirectional (u) and bidirectional (b) reservoirs for all hyper-parameters but for the bias scaling α_{B} in layer 1.

4. RESULTS

Table 2 presents the recognition results of the proposed algorithm with a reservoir size of 20 000. The results show that the bidirectional structure improved the recognition result, because this almost doubled the number of free parameters for the regression. Because of the large dataset, this did not lead into overfitting. The system using two layers with 20 000 reservoir neurons each in the bidirectional configuration is the best performing system for now.

Method		P	R	F
1 layer uni	SJB1	69.05	70.02	69.53
1 layer bi	SJB2	69.13	72.20	70.63
2 layer uni	SJB3	66.43	76.55	71.13
2 layer bi	SJB4	66.86	77.93	71.98

Table 2: P, R and F for different algorithms evaluated on the test set of the MusicNet.

5. ACKNOWLEDGEMENTS

This research was financed by Europäischer Sozialfonds (ESF) and the Free State of Saxony (Application number: 100327771). The parameter optimizations were performed on a Bull Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden.

6. REFERENCES

[1] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In *IFA Proceedings 17*, pages 97–110, 1993.

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

[3] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[4] Azarakhsh Jalalvand, Kris Demuynck, Wesley De Neve, and Jean-Pierre Martens. On the application of reservoir computing networks for noisy image recognition. *Neurocomputing*, 277:237 – 248, 2018. Hierarchical Extreme Learning Machines.

[5] Matthias Mauch and Simon Dixon. PYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659 – 663, May 2014.

[6] Peter Steiner, Simon Stone, Peter Birkholz, and Azarakhsh Jalalvand. Invariances and data augmentation for supervised music transcription. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), unpublished*, 2020.

[7] Simon Stone, Peter Steiner, and Peter Birkholz. A time-warping pitch tracking algorithm considering fast f0 changes. In *Proc. Interspeech 2017*, pages 419–423, 2017.

[8] David Talkin. *A robust algorithm for pitch tracking (RAPT)*, chapter 14, pages 495 – 518. Elsevier Science Inc., New York, NY, USA, 1995.

[9] John Thickstun, Zaid Harchaoui, and Sham M. Kakade. Learning features of music from scratch. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[10] Fabian Triefenbach, Kris Demuynck, and Jean-Pierre Martens. Large vocabulary continuous speech recognition with reservoir-based acoustic models. *IEEE Signal Processing Letters*, 21(3):311 – 315, March 2014.