biomimetics

MDPI

Article

Improving Post-Filtering of Artificial Speech Using Pre-Trained LSTM Neural Networks

Marvin Coto-Jiménez

Escuela de Ingeniería Eléctrica, Universidad de Costa Rica, San José 11501-2060, Costa Rica; marvin.coto@ucr.ac.cr

Received: 14 March 2019; Accepted: 22 May 2019; Published: 28 May 2019

Abstract: Several researchers have contemplated deep learning-based post-filters to increase the quality of statistical parametric speech synthesis, which perform a mapping of the synthetic speech to the natural speech, considering the different parameters separately and trying to reduce the gap between them. The Long Short-term Memory (LSTM) Neural Networks have been applied successfully in this purpose, but there are still many aspects to improve in the results and in the process itself. In this paper, we introduce a new pre-training approach for the LSTM, with the objective of enhancing the quality of the synthesized speech, particularly in the spectrum, in a more efficient manner. Our approach begins with an auto-associative training of one LSTM network, which is used as an initialization for the post-filters. We show the advantages of this initialization for the enhancing of the Mel-Frequency Cepstral parameters of synthetic speech. Results show that the initialization succeeds in achieving better results in enhancing the statistical parametric speech spectrum in most cases when compared to the common random initialization approach of the networks.

Keywords: deep learning; LSTM; machine learning; post-filtering; signal processing; speech synthesis

1. Introduction

Text-to-speech synthesis (TTS) is the technique of generating intelligible speech from a specific text. Applications of TTS have evolved over time, as the quality of the systems has improved to encompass virtual assistants, in-car navigation systems, e-book readers and communicative robots [1]. In the present and future applications, any task that requires the transfer of information between people and machines, or between people with a device as a communication intermediary, becomes a potential area of pertinence [2].

In recent years, TTS systems have progressed from the capacity of producing intelligible speech to the more difficult challenge of generating voices with natural sound, in multiple languages. Despite these trends, there are unresolved obstacles, such as improving the overall quality in the voices generated with concatenated segments of speech and in the past two decades with statistical parametric methods.

The statistical methods have grown in popularity since they arose in the late 1990s [3], particularly those based on Hidden Markov Models (HMMs). HMMs are known for their flexibility in changing speaker characteristics, having a low footprint, and for their capacity to produce advanced features such as average voices [4] and accent modification [5]. Nowadays they are still a preferred technique for several languages and conditions [6].

The main shortcoming of the HMM-based speech synthesis is its quality. It is well known that the generated sequences of parameters from the HMMs are temporally smoothed, producing perceptual differences between synthetic and natural speech. There have been several attempts to improve the quality

of synthesized speech, based on Deep Learning approaches: The first main approach is to substitute the HMM for deep neural networks (DNN) [7–10], learning the map between linguistic specification directly to speech parameters. The second approach is to apply post-filters for the parameters generated by the HMMs [11–13]. The post-filters are usually implemented with DNN, modeling the conditional probability of the acoustic differences between artificial and natural speech.

The main reason to apply DNN is the benefits obtained in many related areas, where the enhancing of signals represents important challenges. For example, to provide better speech recognition in adverse conditions or environments [14,15], and the implementation in low power consumption systems [16].

The high computational cost of training is a significant difficulty in applying some kinds of DNN. For some types of networks, such as Long-Short-term Memory Networks (LSTM), the computational cost has been a shortcoming for the experimentation with more hidden layers or units in the networks [17,18]. With the aim of searching for more efficient ways of training the networks, recent experiences have tested variants of well-known models [19], or the use of extreme learning to explore new conditions [20].

In this work, we explore the benefits of supervised pre-training of LSTM networks as post-filters for spectrum data of synthetic voices generated with HMM, in comparison with the usual random weight initialization. This initialization is performed in the form of an auto-associative network.

1.1. Related Work

Recent experimental results with DNN architectures have been obtained with initialization or training schemes different from the classical feedforward neural networks [21]. For example, unsupervised pre-training that initializes the parameters in a better basin of attraction of the optimization procedure.

In the field of speech technologies, Restricted Boltzmann Machines have been unsupervised initialized and then fine-tuned for speech recognition [22]. The breakthrough to effective training strategies for deep networks came with the algorithms for training deep belief networks, based on a greedy layer-wise unsupervised pre-training followed by supervised fine-tuning [23]. Benefits of the unsupervised pre-training have also been verified in other fields, such as music classification [24] and visual recognition [25]. Semi-supervised techniques applied in similar applications [26] combine at least one stage of unlabeled data to initialize the neural networks.

In artificial speech enhancement, several proposals of neural networks implemented as post-filters have been presented [27], to ensure that the speech parameters of the artificial voices become similar to those of natural speech. A similar approach was presented in [28,29], and enhancement of spectral parameters has been implemented in [30].

In [31], the spectrum features of the synthesized speech are enhanced using networks such as DBNs, and also RBM has been previously studied [29]. More recently, the use of Recurrent Neural Networks (RNNs) was presented in [32], in contrast to standard feedforward networks or models like Deep Belief Nets for post-filtering synthesized speech. The inherent structure of RNNs seems to deal better with the time-dependent nature of the speech signal, which has also been noted in [28]. Previous work using recurrent LSTM networks for the enhancement of the Mel-cepstral coefficients of synthetic voices was recently presented in [18].

In these references, the most common approach is to enhance the spectral components of the synthetic speech, by mapping them to those of the original ones, using diverse deep learning algorithms. A more recent approach that considers a single-stage of multi-stream post-filters has been presented in [33,34], with greater success than the single post-filters based on LSTM.

None of the most recent proposals have made use of pre-training the networks for speech applications. Contrasting with the image classification and voice or music recognition, the post-filtering for synthesized speech is not a classification problem. Here, a regression approach is applied, which has not been

previously tested with pre-training methods. In our proposal, we present a supervised initialization in an auto-associative network for the LSTM autoencoders applied to the enhancement of artificial speech. The supervised initialization, due to the data type, is more natural and provides the networks with a better start for the regression performed in the post-filtering procedure.

The benefits of initialization should also be tested in terms of the quality of synthetic speech, using common measures to assess the spectrum of the signals. None of the previous references with initialization and synthetic speech have implemented such measures. In speech enhancing and related tasks, among the time-domain measures, the most common is the Perceptual Evaluation of Speech Quality (PESQ) [35], which has been presented as an important alternative to costly and time/resource consuming subjective evaluations.

1.2. Problem Statement

In the analysis of natural speech, the trajectories of parameter values (e.g., MFCC, f_0, energy, aperiodic coefficients) exhibit rich modulation characteristics. Conversely, in HMM-based statistical parametric speech synthesis, the speech parameters are overly smoothed due to the statistical modeling and averaging [11,36].

For the task of enhancing the results of HMM-based speech synthesis, we may consider the speech parameters, R_Y, of synthetic utterances as a corrupted or noisy version of the parameters, R_X, of the same original utterances. In a frame-by-frame alignment of synthesized and natural speech, every frame of natural and synthetic speech is parametrized, resulting in a vector:

$$\mathbf{c} = [c_1, c_2, \ldots, c_M] \tag{1}$$

where M is the number of extracted coefficients. c_m can be any kind of parameters, such as f_0, energy or spectrum. For the purpose of this work, we will consider only the MFCC coefficients of both natural and synthetic speech. The analysis of every speech utterance produces a matrix of size $M \times T$ (where T is the number of frames extracted from any utterance) of the form

$$\mathbf{R} = [\mathbf{c}_1^\top, \mathbf{c}_2^\top, \ldots, \mathbf{c}_T^\top] \tag{2}$$

With this notation, let $\mathbf{R_Y}$ and $\mathbf{R_X}$ be the matrices for the synthetic and natural speech respectively (both $M \times T$ size), and $\mathbf{R_W}$ the concatenation of them.

In DNN-based post-filtering, we can enhance the spectral MFCC features of a synthetic voice by estimating a function f_θ from the data, which maps synthetic (noisy) features to natural (clean) features, with some set of weights θ of the network. This can be achieved by minimizing the function [32]:

$$E(\mathbf{R_W}) = ||f_\theta(\mathbf{R_Y}; \mathbf{R_W}) - \mathbf{R_X}||^2 \tag{3}$$

During the training of the DNN-based post-filter, the initial set of weights of the network θ_i are updated each epoch until a stop criteria. In the most traditional approach, a random set of weights became θ_i, and each training epoch updates those weights according to the pair of artificial and natural parameters presented, to a possible set of completely different weights θ_f at the end of the procedure. With this set θ_f it is expected that spectral parameters of any synthetic utterance R_y can be mapped through the neural network to R_x, which presents more natural characteristics.

With the initialization in the form of an auto-associative network, which learns the mapping of identity function from its inputs to its outputs, we began with a set of parameters θ_A that are close of those of θ_f, due to the nature of the regression problem, where $\mathbf{R_Y}$ and $\mathbf{R_X}$ are not completely different and share

some characteristics. In fact, both contain the same set of sounds with the same speech rate but are a pair of natural and synthetic speech.

We pretend to show that θ_A is a better initialization for the LSTM post-filters than θ_R. To show this fact, we propose several experiments with the aim of answering the following questions: (I) Do the benefits of a depend on what data has been used for initialization? (II) Can these benefits be detected with measures of signal quality, and not only with measures of the training procedure itself?

To our knowledge, this is a novel way to initialize, employ and evaluate the LSTM post-filters for synthetic speech. The rest of this chapter is organized as follows: Section 2 provides some details of the LSTM neural networks, of importance in the modeling of MFCC post-filtering. Section 3 describes with detail the Materials and Methods that are part of the proposal. Section 4 presents the Results, whilst Section 5 shows the Discussion of the results. Finally, conclusions are given in Section 6.

2. Long Short-Term Memory Recurrent Neural Networks

Among the many new algorithms developed to improve some tasks related to speech, such as speech synthesis recognition, several groups of researchers have experimented with the use of DNN, with encouraging results. Deep learning, based on several kinds of neural networks with many hidden layers, have achieved important results in many machine learning and pattern recognition problems. The disadvantage of using such networks is that they cannot directly model the dependent nature of sequential parameters, something which is desirable to imitate human speech production. It has been suggested that one way to solve this problem is to include RNN [37,38] in which there is feedback from some of the neurons in the network, either backward or to themselves, forming a kind of memory that retains information about previous states.

An extended kind of RNN, which can store information over long or short time intervals, has been presented in [39] and is called LSTM. Recently, LSTM was successfully used in speech recognition as well as in other applications to speech recognition [17,40,41] and classification [42,43]. The storage and use of long-term and short-term information within the network are potentially significant for many applications, including speech processing, non-Markovian control, and music composition [39].

In an LSTM network with several layers of units with memory, output vector sequences $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ are computed from input vector sequences $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ and hidden vector sequences $\mathbf{h} = (h_1, h_2, \ldots, h_T)$, iterating Equations (4) and 5 from 1 to T [37]:

$$h_t = \mathcal{H}\left(\mathbf{W}_{xh}x_t + \mathbf{W}_{hh}h_{t-1} + b_h\right) \tag{4}$$

$$y_t = \mathbf{W}_{hy}h_t + b_y \tag{5}$$

where $\mathbf{W}_{ij}$ is the weight matrix between layer i and j, b_k is the bias vector for layer k and $\mathcal{H}$ is the activation function for hidden nodes, usually a sigmoid function $f : \mathbb{R} \to \mathbb{R}, f(t) = \frac{1}{1+e^{-t}}$ or a hyperbolic tangent function.

Each cell in the hidden layers of an LSTM has some extra gates to store values, in comparison to other RNN networks: an input gate, forget gate, output gate and cell activation. With the proper combination of these gates, the values propagated through the network can be stored in the long or short term, and easily released to units of the same layer or next layers of the network to improve the capacity of the network with information of past states. The gates of a typical LSTM network are implemented following the equations:

$$i_t = \sigma\left(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}h_{t-1} + \mathbf{W}_{ci}c_{t-1} + b_i\right) \tag{6}$$

$$f_t = \sigma\left(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}h_{t-1} + \mathbf{W}_{cf}c_{t-1} + b_f\right) \tag{7}$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(\mathbf{W}_{xc} x_t + \mathbf{W}_{hc} h_{t-1} + b_c\right) \quad (8)$$

$$o_t = \sigma\left(\mathbf{W}_{xo} x_t + \mathbf{W}_{ho} h_{t-1} + \mathbf{W}_{co} c_t + b_o\right) \quad (9)$$

$$h_t = o_t \tanh\left(c_t\right) \quad (10)$$

where σ is the sigmoid function, i is the input gate activation function, f the forget gate activation function, o is the output gate activation function, and c the cell activation function. $\mathbf{W}_{mn}$ are the weight matrices from each cell to gate vector. h is the output of the unit.

The combination of gates allows an LSTM neural network to decide when to keep or override information in the memory cell, when to access memory cell and when to prevent other units from being perturbed by the memory cell value [39]. More details on the training procedures and the applications of LSTM networks can be found in [44]. The storage of the previous states in the memory of the LSTM can benefit the improvement of the spectrum of speech signals due to their time-dependent nature of past states, which otherwise could not be stored for use in other types of non-recurring neural networks.

3. Materials and Methods

The literature related to statistical parametric speech synthesis produced using HMM, have repeatedly reported notable differences with the artificial and the original voices (e.g., [45–47]). As described in previous sections, it is possible to reduce the gap between natural and artificial voices by learning a function that maps them directly from the data [29]. In our proposal, we use aligned utterances from natural and synthetic voices produced by the HTS system [48] to establish a correspondence between each frame. Then, LSTM post-filters are trained to map MFCC parameters of synthetic speech into those corresponding to natural speech. For this purpose, our interest is to show the advantage of initializing the weights of the networks in terms of the quality of the results. We initialize the network to establish this advantage, in the following ways:

- Randomly: All the weights have random numbers at the first epoch of training. This initialization method is the most common usage DNN-based post-filtering of artificial speech, and we take it as the base system.
- Initialized-1: The weights are initialized using an auto-associative network (ANN). ANN is a neural network whose input and target vectors are the same [49]. From this definition, it is important to explore whether the ANN can be set from clean and artificial speech spectrum data. Initialized-1 refers to the case of clean speech.
- Initialized-2: The weights of the network are initialized in the form of an ANN trained with artificial speech spectrum vectors.

The ANN was trained with the following procedure: An LSTM network with the same architecture of the post-filters (39 units as inputs and 39 as outputs for the enhancing of MFCC) was trained with the same data at the input and the output in each frame. This way, the network learns the identity function between its inputs and its outputs. After training, the weights of the ANN became the initialized weights of the corresponding post-filters. The indication "1" means the usage of clean data during the pre-training of the LSTM network.

In every case after the pre-training process, the inputs to the LSTM network correspond to the 39 MFCC parameters of each frame for the sentences spoken using the HMM-based voice, while the output corresponds to the MFCC parameters given by the natural voice for the same sentence. Hence, each LSTM post-filter network attempts to solve the regression problem of transforming the values of speech produced by the HMM-based system into those of the natural voices. The experiments allow comparing the two types of initialization: the traditional random approach and the auto-associative proposal (in two cases,

using natural or artificial coefficients for the training of the ANN), where the post-filter needs to refine the weights after the described process.

3.1. Corpus Description

For the experimentation, we used the CMU Arctic databases, constructed at the Language Technologies Institute at Carnegie Mellon University [50]. They are phonetically balanced and contain the speech of five US English speakers. The databases were designed for research in unit selection speech synthesis, commonly applied also to HMM-based speech synthesis, and consist of around 1150 utterances selected from out-of-copyright texts from Project Gutenberg.

The databases include male and female speakers. A detailed report on the structure and content of the database and the recording conditions is available in the Language Technologies Institute Tech Report CMU-LTI-03-177 18. The five voices that were chosen here for the experiments are identified by a three letters: BDL (male), CLB (female), JMK (male), RMS (male) and SLT (female).

3.2. Feature Extraction

The audio files of the database were downsampled to 16 kHz, in order to extract the parameters using the Ahocoder system. In this system, the fundamental frequency f_0^k (zero-valued if invoiced), 39 MFCC, plus an energy coefficient are extracted from each frame. Hence each frame is represented by a 41st dimensional vector $V_k = \left[f_0^k, e^k, mfcc_k^1, \ldots, mfcc_k^{39}\right]$. For this work, we only kept the 39 MFCC of the parametrization, whilst the remaining parameters were leaving unchanged from the HMM-based voice. Further, an audio waveform can be synthesized from a similar set of parameters using the system (Ahodecoder). Details on the parameter extraction and waveform regeneration of this system can be found in [51].

3.3. Experiments

Each of the five voices was parameterized, and the resulting set of vectors was divided into training, validation, and testing sets. The amount of data available for each voice is shown in Table 1. Despite all voices uttering the same phrases, the length differences are due to variations in the speaker's rate. The test set consists of 50 utterances. The stop criteria of the training process were established in terms of a maximum number of epochs (500) or 25 epochs since the last lower sse value for the validation set.

Table 1. Amount of data (vectors) available for each voice in the databases.

Voice	Gender/Accent	Total	Train	Validation	Test
BDL	(M) US-English	676,554	473,588	135,311	67,655
SLT	(F) US-English	677,970	474,579	135,594	67,797
CLB	(F) US-English	769,161	538,413	153,832	76,916
RMS	(M) US-English	793,067	555,147	158,613	79,307
JMK	(M) US-English	635,503	541,856	62,135	31,512

(a) JMK voice in US-English was produced from a Canadian English speaker.

The AAN initialization was performed with the parameters of all the voices, i.e., there are 10 different cases:

- Five cases where the set of ANN was trained using clean MFCC coefficients (one for each voice). In the results, these cases are identified with the superscript 1.
- Five cases where the set of ANN was trained using MFCC coefficients from the HMM-based voices (one for each voice). In the results, these cases are identified with the superscript 2.

Additionally, to establish a comparison to the traditional random initialization, we performed the post-filtering with three LSTM networks trained independently with random weight initialization.

The LSTM post-filter architecture was defined after a process of trial and error, using the Currennt system [52]. The final selection of three hidden layers, with 150, 100 and 150 units in each one respectively was considered after producing the best results in the first set of experiments, and also having manageable training time for the 25 LSTM networks used in the present work (10 for the initialization of both cases of ANN and 15 for the three-time random initialization).

The training procedure was accelerated by an NVIDIA GPU system and took a mean time of 12 h to train each LSTM. For the whole set of experiments, there was a running time of more than 12 days.

In order to determine the improvement in the efficiency of the supervised pre-training, we use the following objective measures:

- Number of epochs: The time taken to train a neural network is directly associated with the amount of epochs in training. Each epoch consists in a forward and backward pass of data, and the updating of the weights. The lesser epochs it takes to train a network, the less time is needed, so the procedure is more effective.
- sse: This is a common measure for the error in the validation and test sets during training. Is defined as:

$$\text{sse}(\theta) = \sum_{n=1}^{T} (\mathbf{c_x} - \hat{\mathbf{c}_x})^2 \tag{11}$$

$$= \sum_{n=1}^{T} (\mathbf{c_x} - f(\mathbf{c_x}))^2 \tag{12}$$

  where $\mathbf{c_x}$ is, $\hat{\mathbf{c}}_\mathbf{x}$ is T the number of frames, and f the function the networks perform between its inputs and its outputs. A lower value of sse means that the network is producing outputs more closely related to the objective parameters (those of the natural voice).

Additionally, to verify whether the results represent improvement in the quality of the speech, we incorporate the following measure:

- PESQ: This measure uses a psychoacoustic model to predict the subjective quality of speech. This measure is defined in the ITU-T recommendation P.862.ITU. Results are given in interval $[0.5, 4.5]$, where 4.5 corresponds to a perfect reconstruction of the signal.

  PESQ is computed as [53]:

$$\text{PESQ} = a_0 + a_1 D_{ind} + a_2 A_{ind} \tag{13}$$

  where the D_{ind} is the average disturbance and A_{ind} the asymmetrical disturbance. The a_k are chosen to optimize PESQ in measuring speech distortion, noise distortion and overall quality.

The results and analysis are shown in the following sections.

4. Results

The results are divided into two sections: The first part emphasizes the improvement achieved in the efficiency of the LSTM networks, while the second part shows the results of the metrics used to determine the improvement in the quality of the voices due to the post-filter process. In all cases, the ANN training was stopped by the criteria of a maximum number of epochs (500).

4.1. Training Efficiency

In terms of the efficiency, the lower number of epochs is associated with less time in the training procedure, while the lower value of sse means a better performance of the network for the task of regression from the synthetic parameters to the natural ones.

Table 2 presents the results of both the number of epochs and sse for the five voices. The values reported in this table correspond to the best case of the three random initialization and the initialization of the network with natural data (ANN-1). In the SLT voice, the MFCC post-filter required 95 fewer epochs to train with the auto-associative initialization (29% less time), with a benefit of lowering the sse from 290.00 to 276.33.

These results show a more efficient and effective way of training with the supervised initialization. The rest of the voices present improvements in the training time in 43% (BDL voice), 32% (CLB voice), 25% (RMS voice) and 14% (JMK voice). The sse values present improvements in all the cases where the ANN initialization was applied.

Figure 1 shows the evolution of sse in each training epoch, and illustrates how the auto-associative initialization reaches lower sse with fewer epochs. It can be noticed also how the first epochs represent the greater differences during the training process.

Table 2. Comparison of the results for the test set during training the Long Short-term Memory (LSTM) networks to enhance MFCC.

Random initialization									
SLT		BDL		CLB		RMS		JMK	
Epochs	sse	Epochs	sse	Epochs	sse	Epochs	sse	Epochs	sse
327	290.00	236	378.58	198	362.56	196	382.80	232	352.36
ANN initilization 1									
SLT		BDL		CLB		RMS		JMK	
Epochs	sse	Epochs	sse	Epochs	sse	Epochs	sse	Epochs	sse
232	276.33	134	364.28	135	350.92	147	368.07	200	341.72

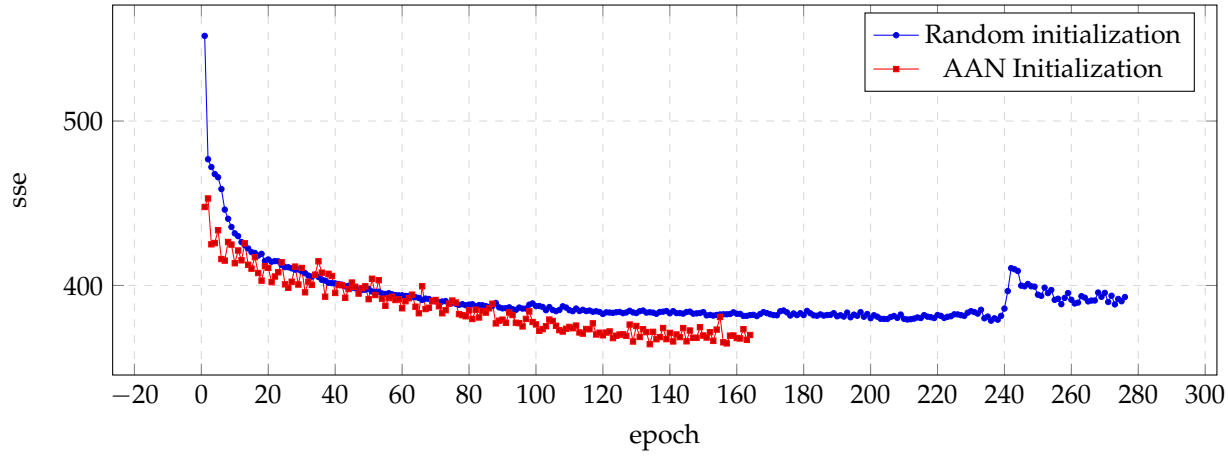

Figure 1. Evolution of the sse value for the validation set during the training process of the BDL voice.

The results of the sse measure for the CLB voice are shown in Figure 2. This figure illustrates how the auto-associative initialization reaches lower sse values with fewer epochs. During the first part of the process (about 40 epochs), the benefits are not noticeable yet, but at the end of the training, they are quite remarkable.

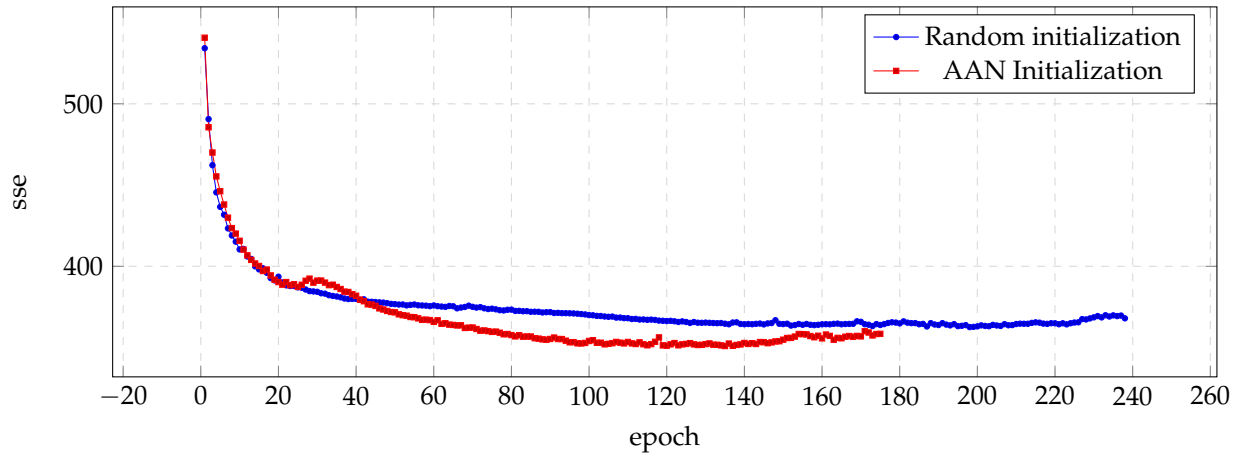

Figure 2. Evolution of the sse value for the validation set during the training process of the CLB voice.

Similar results are presented in Figure 3 (JMK voice) and Figure 4 (RMS voice), with most significant improvements in the case of the JMK voice.

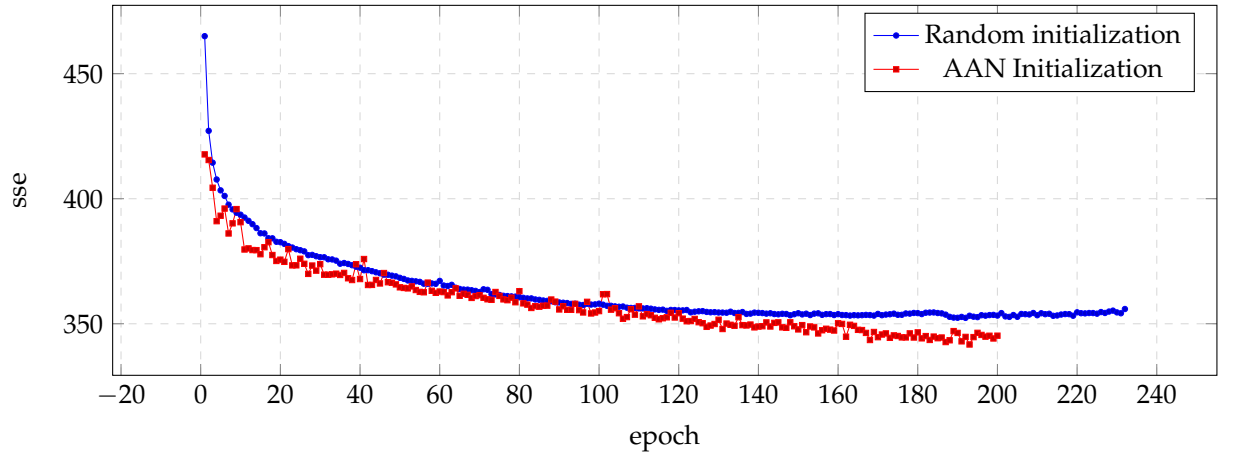

Figure 3. Evolution of the sse value for the validation set during the training process of the JMK voice.

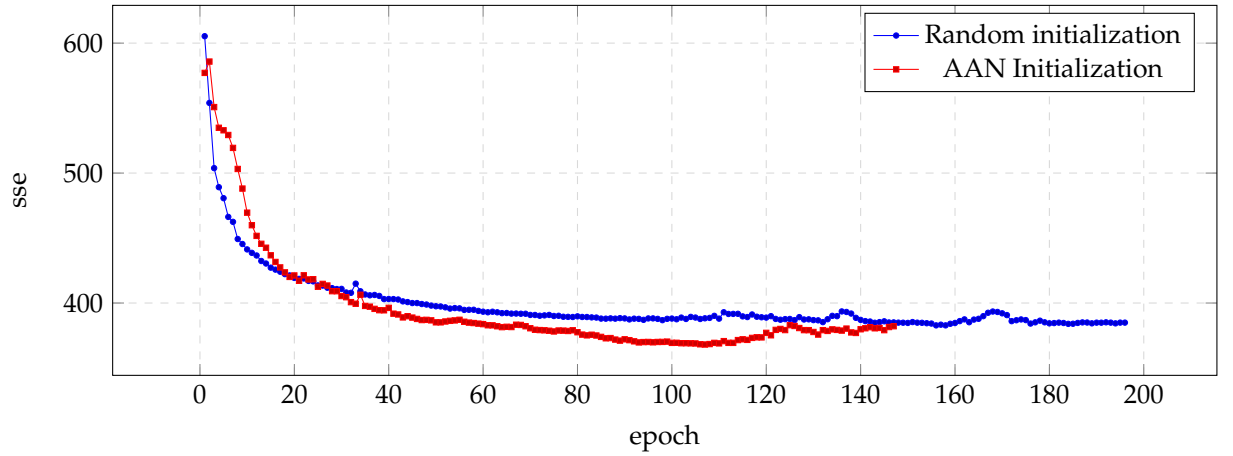

Figure 4. Evolution of the sse value for the validation set during the training process of the RMS voice.

Finally, one of the most significant results is shown in Figure 5, where the evolution of the sse value for the validation set is noticeable during the complete process.

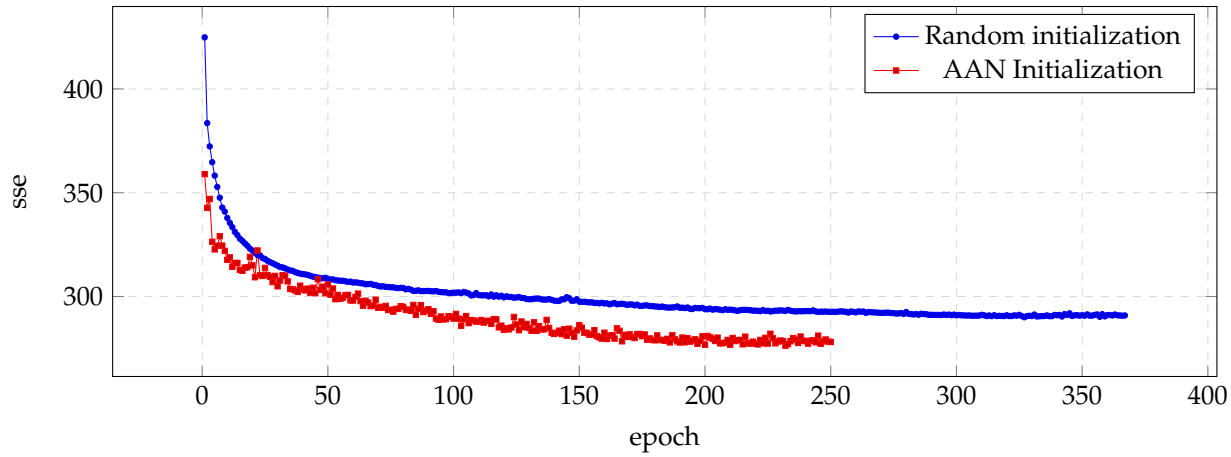

Figure 5. Evolution of the sse value for the validation set during the training process of the SLT voice.

4.2. Quality of the Results

Apart from the improvements in the efficiency in terms of objective measures of training time and sse, the convenience of applying the initialization of the LSTM post-filters with the ANN should be tested in terms of the quality of the post-filtering process, which pretends to improve the spectrum of synthetic voices.

Since initialization requires a previous step of training, it is important to establish the independence of the results of the data used in the process. For this reason, in this part of the results, the PESQ was applied to the 50 utterances of the test set considering the case where initialization was performed using natural or synthetic speech parameters. In addition, it is important to compare the results to the random initialization that is considered the base case.

Table 3 presents the result for the case of BDL voice. Here, the best mean value of the PESQ was obtained with the network pre-trained with synthetic parameters of the RMS voice.

Table 3. Mean PESQ Results for BDL voice. Higher values represent better results. * is the best result. The superscript 1 means that the LSTM post-filter was pre-trained as ANN using natural parameters, while the superscript 2 means the same procedure applied with synthetic parameters.

Random-Worst	**Random-Best**	**BDL** ^{1}	**CLB** ^{1}	**JMK** ^{1}	**RMS** ^{1}	**SLT** ^{1}
1.45	1.46	1.45	1.46	1.45	1.44	1.46
		BDL ^{2}	**CLB** ^{2}	**JMK** ^{2}	**RMS** ^{2}	**SLT** ^{2}
		1.43	1.45	1.47	1.49 *	1.47

Table 4 shows the result for the case of CLB voice. Here, the best mean value of the PESQ was obtained with the network pre-trained with synthetic parameters of the RMS voice. The best results were achieved with the initialization of the post-filter using natural parameters of the RMS and SLT voices and synthetic parameters of the RMS voice.

Table 4. Mean PESQ Results for CLB voice. Higher values represent better results. * is the best result. The superscript 1 means that the LSTM post-filter was pre-trained as ANN using natural parameters, while the superscript 2 means the same procedure applied with synthetic parameters.

Random-Worst	Random-Best	BDL [1]	CLB [1]	JMK [1]	RMS [1]	SLT [1]
1.16	1.19	1.18	1.20	1.20	1.23*	1.23 *
		BDL [2]	**CLB [2]**	**JMK [2]**	**RMS [2]**	**SLT [2]**
		1.20	1.20	1.22	1.23*	1.18

The PESQ results of the JMK voice are presented in Table 5. The best result was obtained with the ANN initialization performed with natural parameters of the BDL voice, and the second to best with artificial parameters of the JMK voice.

Table 5. Mean PESQ Results for JMK voice. Higher values represent better results.

Random-Worst	Random-Best	BDL [1]	CLB [1]	JMK [1]	RMS [1]	SLT [1]
1.51	1.56	1.59 *	1.53	1.56	1.56	1.54
		BDL [2]	**CLB [2]**	**JMK [2]**	**RMS [2]**	**SLT [2]**
		1.55	1.55	1.58	1.55	1.55

The only case where the ANN initialization of the LSTM post-filter did not achieve a better result than the random initialization, was the RMS voice. These results are shown in Table 6.

Finally, the results for the SLT voice are presented in Table 7. The best results were obtained with the ANN initialization performed with data from the JMK voice.

Table 6. Mean PESQ results for RMS voice. Higher values represent better results.

Random-Worst	Random-Best	BDL [1]	CLB [1]	JMK [1]	RMS [1]	SLT [1]
1.70	1.72 *	1.71	1.71	1.71	1.68	1.71
		BDL [2]	**CLB [2]**	**JMK [2]**	**RMS [2]**	**SLT [2]**
		1.71	1.69	1.70	1.70	1.70

Table 7. Mean PESQ Results for SLT voice. Higher values represent better results.

Random-Worst	Random-Best	BDL [1]	CLB [1]	JMK [1]	RMS [1]	SLT [1]
0.95	0.97	0.95	0.97	0.98 *	0.95	0.96
		BDL [2]	**CLB [2]**	**JMK [2]**	**RMS [2]**	**SLT [2]**
		0.97	0.97	0.98 *	0.97	0.94

To illustrate the effect of the initialization in the results, Figures 6 and 7 show the evolution of the first MFCC coefficient of the BDL and the fifth of the SLT voice, respectively. In all cases, the figures present the result of the best case according to PESQ measure. It can be seen that the results of the post-filters initialized with ANN tend to present values closer to those of the natural voice.

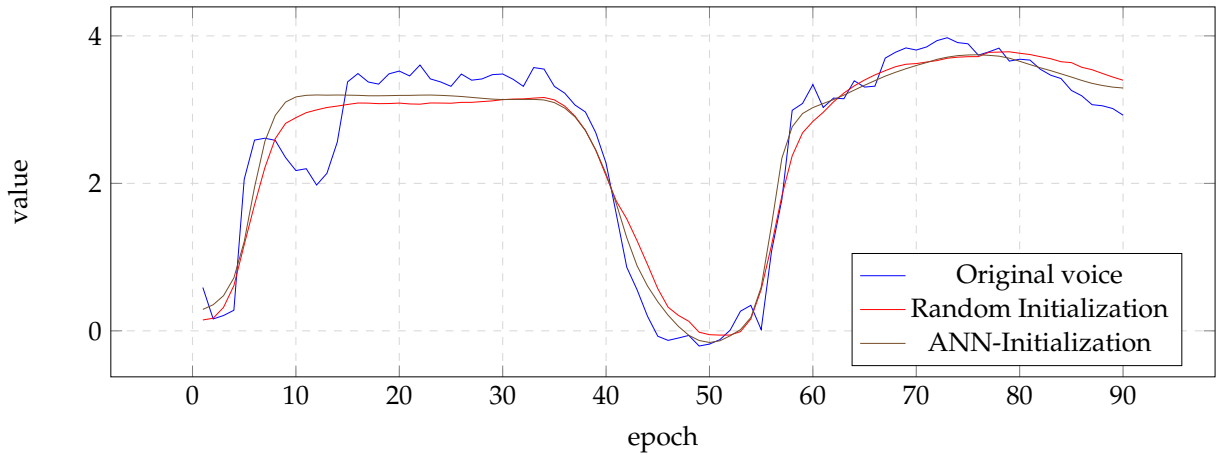

Figure 6. First MFCC for the BDL voice.

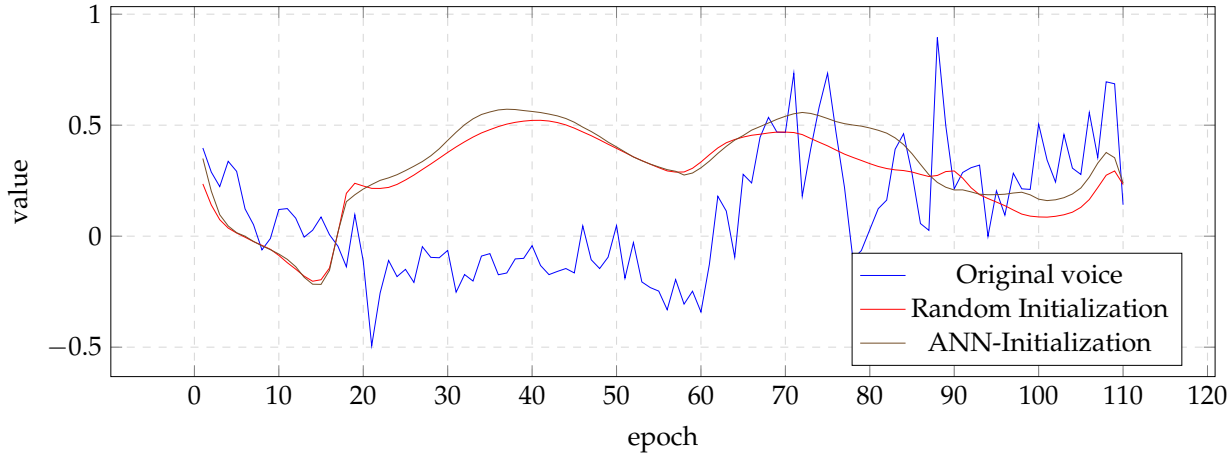

Figure 7. Sample contour of the fifth MFCC for the SLT voice.

5. Discussion

The advantage of proposed ANN-initialization lies on fewer training epochs required to achieve even better sse values than the usual random initialization. All the cases analyzed allow to confirm this benefit of the proposal.

As shown in Table 2, the LSTM post-filters presented significant variations in the number of epochs required for training. These differences can be partially explained by the fact that the quality of synthetic speech relies on the characteristics of the database. This fact means it is expected in the results noticeable differences between the natural and synthetic voices, which can be represented as a bigger or lower gap to map between them in the post-filters.

According to the results in terms of the PESQ measure, neither the number of epochs nor the sse values can be used as a substitute for quality measures of the results. Table 6 is the main evidence among the experiments, which reflects that even though the ANN initialization lowered sse values with fewer epochs, it failed in presenting a better PESQ result than the random initialization. Given that quality measurement is being carried out on the complete speech wave, it is important to emphasize that only the MFCC coefficients are being processed, while the energy, aperiodic coefficients, and f_0 remained the same, and also have a significant effect on the measurements quality.

Regarding the independence of the data utilized in the pre-training, the results of the BDL voice (Table 3) show significant differences between them. For both the initialization with natural and synthetic data, the ANN initialization shows the best and worst results compared to the random initialization. The CLB voice (Table 4) presents the case where the initialization produces better results than the random initialization in most cases. In particular, it is important to remark that the initialization with parameters of the same CLB voice produces the best results of the three random initializations.

Similar results for the two types of initialization are shown for the JMK voice in Table 5. Here, most of the values are better than the worst case of random initialization, and in all cases, the results were better than the worst case of random initialization. In particular, the initialization with data of the same JMK voice results in better PESQ values than the random base case.

For the case of RMS voice, only two cases of initialization produce values lower than the worst case of random initialization. The case of ANN-1 pre-training with the parameters of the same RMS voice can be considered as an exception among the results because they present the lower value.

The SLT voice shows the benefit of the ANN pre-training proposal of the LSTM post-filters, because all the networks, except for one case, present better results and the worst random initialization, and in most cases are equal to or better than the best random case.

It can be noticed, by comparing all the cases studied in this paper, that only using the initialization with JMK synthetic parameters of the LSTM networks, better results than all the random procedures were obtained in four of the five cases. This fact can support one of the most significant results of this work: that the benefits of the pre-training of LSTM networks for the post-filtering of HMM-based voices depend on the quality of the ANN pre-training.

6. Conclusions

In this paper, we have presented a proposal of initialization of LSTM post-filters for the enhancement of statistical parametric artificial voices, based on an auto-associative network. We conducted a comparison using parameters of five voices, both male and female, and two well-known measures for the efficiency of the training neural networks and the most relevant measure for the quality of the speech.

The main assumption for applying the initialization was that the regression performed from the artificial to natural parameters of the voice preserves many characteristics from the input to the output, so the usual random initialization of the network is not the most convenient state to reach the set of weights that best suit the mapping. That is why an auto-associative neural network, which learns the identity function in a supervised way during the pre-training stage, is a better approximation to the desired mapping.

The results show that the auto-associative initialization has benefits in terms of less training epochs and less sum of squared errors for all the voices. The proposed initialization seems independent of the type of data (natural or synthetic) used during the pre-training stage. However, better results tend to be obtained from the initialization with synthetic data.

Due to the quality of synthetic voices relying on parameters other than the MFCC, future work should include extending this proposal of initialization of LSTM networks for the rest of the parameters of synthetic speech.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the University of Costa Rica (UCR), Project No. 322-B9-105.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AAN	Auto-associative Network
DNN	linear dichroism
HMM	Three letter acronym
LSTM	Multidisciplinary Digital Publishing Institute
PESQ	Directory of open access journals
RBM	Restricted Boltzman Machine
RNN	Recurrent Neural Network
TTS	Text-to-Speech Synthesis
f_0	Fundamental frequency

References

1. Tokuda, K.; Nankaku, Y.; Toda, T.; Zen, H.; Yamagishi, J.; Oura, K. Speech synthesis based on hidden Markov models. *Proc. IEEE* **2013**, *101*, 1234–1252. [CrossRef]
2. Holmes, W.; Holmes, J. *Speech Synthesis and Recognition*; CRC Press: Boca Raton, FL, USA, 2001; pp. 93–107.
3. Yoshimura, T.; Tokuda, K.; Masuko, T.; Kobayashi, T.; Kitamura, T. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In Proceedings of the Sixth European Conference on Speech Communication and Technology, Budapest, Hungary, 5–9 September 1999; pp. 2347–2350.
4. Tamura, M.; Masuko, T.; Tokuda, K.; Kobayashi, T. Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR. *Proc. IEEE Acoust. Speech Signal Process.* **2001**, *2*, 805–808.
5. Coto-Jiménez, M.; Goddard-Close, J. Hidden Markov Models for Artificial Voice Production and Accent Modification. In Proceedings of the Ibero-American Conference on Artificial Intelligence, San Jose, Costa Rica, 22–25 November 2016; pp. 415–426.
6. Biagetti, G.; Crippa, P.; Falaschetti, L.; Turchetti, C. HMM speech synthesis based on MDCT representation. *Int. J. Speech Technol.* **2018**, *21*, 1045–1055. [CrossRef]
7. Ze, H.; Senior, A.; Schuster, M. Statistical parametric speech synthesis using deep neural networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013.
8. Wu, Z.; Valentini-Botinhao, C.; Watts, O.; King, S. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 20–25 March 2016.
9. Zen, H.; Senior, A. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014.
10. Wang, X.; Takaki, S.; Yamagishi, J. Investigating very deep highway networks for parametric speech synthesis. *Speech Commun.* **2018**, *96*, 1–9. [CrossRef]
11. Chen, L.H.; Raitio, T. DNN-based stochastic postfilter for HMM-based speech synthesis. In Proceedings of the INTERSPEECH, Singapore, 14–18 September 2014.
12. Okamoto, T.; Tachibana, K.; Toda, T.; Shiga, Y.; Kawai, H. Deep neural network-based power spectrum reconstruction to improve quality of vocoded speech with limited acoustic parameters. *Acoust. Sci. Technol.* **2018**, *39*, 163–166. [CrossRef]
13. Saito, Y.; Takamichi, S.; Saruwatari, H. Statistical Parametric Speech Synthesis Incorporating Generative Adversarial Networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 84–96. [CrossRef]
14. Siniscalchi, S.M.; Valerio, M.S. Adaptation to new microphones using artificial neural networks with trainable activation functions. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 1959–1965. [CrossRef] [PubMed]

15. Hung, J.; Jung-Shan, L.; Po-Jen, W. Employing Robust Principal Component Analysis for Noise-Robust Speech Feature Extraction in Automatic Speech Recognition with the Structure of a Deep Neural Network. *Appl. Syst. Innov.* **2018**, *1*, 28. [CrossRef]
16. Pal Chowdhury, A.; Kulkarni, P.; Nazm Bojnordi, M. MB-CNN: Memristive Binary Convolutional Neural Networks for Embedded Mobile Devices. *J. Low Power Electron. Appl.* **2018**, *8*, 38. [CrossRef]
17. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In Proceedings of the International Conference on Artificial Neural Networks, Killarney, Ireland, 12–16 July 2015.
18. Coto-Jiménez, M.; Goddard-Close, J.; Martínez-Licona, F.M. Improving automatic speech recognition containing additive noise using deep denoising autoencoders of LSTM networks. In Proceedings of the International Conference on Speech and Computer, Budapest, Hungary, 23–27 August 2016.
19. Fei, H.; Fengyun, T. Bidirectional Grid Long Short-Term Memory (BiGridLSTM): A Method to Address Context-Sensitivity and Vanishing Gradient. *Algorithms* **2018**, *11*, 172. [CrossRef]
20. Salerno, V.; Rabbeni, G. An extreme learning machine approach to effective energy disaggregation. *Electronics* **2018**, *7*, 235. [CrossRef]
21. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010.
22. Dahl, G.E.; Yu, D.; Deng, L.; Acero, A. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 30–42. [CrossRef]
23. Erhan, D.; Bengio, Y.; Courville, A.; Manzagol, P.A.; Vincent, P.; Bengio, S. Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* **2010**, *11*, 625–660.
24. Van Den Oord, A.; Dieleman, S.; Schrauwen, B. Transfer learning by supervised pre-training for audio-based music classification. In Proceedings of the Conference of the International Society for Music Information Retrieval), Taipei, Taiwan, 27–31 October 2014.
25. Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
26. Vesely, K.; Hannemann, M.; Burget, L. Semi-supervised training of deep neural networks. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Olomouc, Czech Republic, 8–12 December 2013.
27. Takamichi, S.; Toda, T.; Black, A.W.; Nakamura, S. Modified post-filter to recover modulation spectrum for HMM-based speech synthesis. In Proceedings of the IEEE Global Signal and Information Processing Conference, Atlanta, GA, USA, 3–5 December 2014. [CrossRef]
28. Takamichi, S.; Toda, T.; Black, A.W.; Neubig, G.; Sakti, S.; Nakamura, S. Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. *Proc. IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 755–767. [CrossRef]
29. Chen, L.H.; Raitio, T.; Valentini-Botinhao, C.; Ling, Z.H.; Yamagishi, J. A deep generative architecture for postfiltering in statistical parametric speech synthesis. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2015**, *23*, 2003–2014. [CrossRef]
30. Takamichi, S.; Toda, T.; Neubig, G.; Sakti, S.; Nakamura, S. A postfilter to modify the modulation spectrum in HMM-based speech synthesis. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014. [CrossRef]
31. Nakashika, T.; Takashima, T.; Takiguchi, T.; Ariki, Y. Voice conversion in high-order eigen space using deep belief nets. In Proceedings of the INTERSPEECH, Lyon, France, 25–29 August 2013.
32. Muthukumar, P.K.; Black, A.W. Recurrent Neural Network Postfilters for Statistical Parametric Speech Synthesis. *arXiv* **2016**, arXiv:1601.07215. .
33. Coto-Jiménez, M.; Goddard-Close, J. LSTM Deep Neural Networks Postfiltering for Enhancing Synthetic Voices. *Int. J. Pattern Recognit. Artif. Intell.* **2018**, *32*. [CrossRef]

34. Coto-Jiménez, M.; Goddard-Close, J. LSTM Deep Neural Networks Postfiltering for Improving the Quality of Synthetic Voices. In Proceedings of the Mexican Conference on Pattern Recognition, Guanajuato, Mexico, 22–25 June 2016.
35. Norrenbrock, C.R.; Hinterleitner, F.; Heute, U.; Möller, S. Quality prediction of synthesized speech based on perceptual quality dimensions. *Speech Commun.* **2015**, *66*, 17–35. [CrossRef]
36. Nguyen, G.; Phung, T. Reducing over-smoothness in HMM-based speech synthesis using exemplar-based voice conversion. *EURASIP J. Audio Speech Music Process.* **2017**, *1*, 14. [CrossRef]
37. Fan, Y.; Qian, Y.; Xie, F.L.; Soong, F.K. TTS synthesis with bidirectional LSTM based recurrent neural networks. In Proceedings of the 15th Annual Conference of the International Speech Communication Association, Singapore, 14–18 September 2014.
38. Zen, H.; Sak, H. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Queensland, Australia, 19–24 April 2015. [CrossRef]
39. Sepp, H.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780.
40. Graves, A.; Jaitly, N.; Mohamed, A. Hybrid speech recognition with deep bidirectional LSTM. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278. [CrossRef]
41. Chiu, C.; Sainath, T.N.; Wu, Y.; Prabhavalkar, R.; Nguyen, P.; Chen, Z.; Kannan, A.; Weiss, R.J.; Rao, K.; Gonina, E. State-of-the-art speech recognition with sequence-to-sequence models. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018.
42. Harati, S.; Crowell, A.; Mayberg, H.; Nemati, S. Depression Severity Classification from Speech Emotion. In Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018.
43. Wang, J.H.; Liu, T.W.; Luo, X.; Wang, L. An LSTM Approach to Short Text Sentiment Classification with Word Embeddings. In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018), Hanoi, Vietnam, 18–24 March 2018.
44. Gers, F.A.; Schraudolph, N.; Schmidhuber, J. Learning precise timing with LSTM recurrent networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143.
45. Toda, T.; Tokuda, K. A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Trans. Inf. Syst.* **2007**, *90*, 816–824. [CrossRef]
46. Stan, A.; Yamagishi, J.; King, S.; Aylett, M. The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate. *Speech Commun.* **2011**, *53*, 442–450. [CrossRef]
47. Wu, Y.; King, S.; Tokuda, K. Cross-lingual speaker adaptation for HMM-based speech synthesis. In Proceedings of the IEEE 6th International Symposium on Chinese Spoken Language Processing, Kunming, China, 16–19 December 2008.
48. The HTS Group. HMM/DNN-Based Speech Synthesis System (HTS). Available online: http://hts.sp.nitech.ac.jp/ (accessed on 20 February 2019).
49. Baek, J.; Cho, S. Bankruptcy prediction for credit risk using an auto-associative neural network in Korean firms. In Proceedings of the IEEE International Conference on Computational Intelligence for Financial Engineering, Hong Kong, China, 20–23 March 2003.
50. Kominek, J.; Black, A.W. The CMU Arctic speech databases. In Proceedings of the Fifth ISCA Workshop on Speech Synthesis, Pittsburgh, PA, USA, 14–16 June 2004.
51. Erro, D.; Sainz, I.; Saratxaga, I.; Navas, E.; Hernáez, I. MFCC+F0 extraction and waveform reconstruction using HNM: Preliminary results in an HMM-based synthesizer. In Proceedings of the VI Jornadas en Tecnologia del Habla & II Iberian SLTech (FALA) Workshop, Vigo, Spain, 10–12 November 2010; pp. 29–32.

52. Weninger, F.; Bergmann, J.; Schuller, B. Introducing CURRENNT—The Munich Open-Source CUDA RecurREnt Neural Network Toolkit. *J. Mach. Learn. Res.* **2014**, *16*, 547–551.
53. Rix, A.W.; Beerends, J.G.; Hollier, M.P.; Hekstra, A.P. Perceptual evaluation of speech quality (PESQ)—A new method for speech quality assessment of telephone networks and codecs. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, UT, USA, 7–11 May 2001.