2021

Deep learning versus spectral techniques for frequency estimation of single tones: Reduced complexity for software-defined radio and iot sensor communications

Hind R. Almayyali

Zahir M. Hussain
Edith Cowan University

MDPI

Article

Deep Learning versus Spectral Techniques for Frequency Estimation of Single Tones: Reduced Complexity for Software-Defined Radio and IoT Sensor Communications

Hind R. Almayyali [1] and Zahir M. Hussain [1,2,*]

1   Computer Science and Mathematics, University of Kufa, Najaf 54001, Iraq; hindalmayyali@gmail.com
2   School of Engineering, Edith Cowan University, Joondalup, WA 6027, Australia
*   Correspondence: z.hussain@ecu.edu.au or zmhussain@ieee.org

Abstract: Despite the increasing role of machine learning in various fields, very few works considered artificial intelligence for frequency estimation (FE). This work presents comprehensive analysis of a deep-learning (DL) approach for frequency estimation of single tones. A DL network with two layers having a few nodes can estimate frequency more accurately than well-known classical techniques can. While filling the gap in the existing literature, the study is comprehensive, analyzing errors under different signal-to-noise ratios (SNRs), numbers of nodes, and numbers of input samples under missing SNR information. DL-based FE is not significantly affected by SNR bias or number of nodes. A DL-based approach can properly work using a minimal number of input nodes N at which classical methods fail. DL could use as few as two layers while having two or three nodes for each, with the complexity of $O\{N\}$ compared with discrete Fourier transform (DFT)-based FE with $O\{N\log_2(N)\}$ complexity. Furthermore, less N is required for DL. Therefore, DL can significantly reduce FE complexity, memory cost, and power consumption, which is attractive for resource-limited systems such as some Internet of Things (IoT) sensor applications. Reduced complexity also opens the door for hardware-efficient implementation using short-word-length (SWL) or time-efficient software-defined radio (SDR) communications.

Keywords: frequency estimation; deep-learning (DL); sensors; Internet of Things (IoT); short word length (SWL); software-defined radio (SDR); parallel-computing fast Fourier transform (FFT); low-power; low-cost; biomedical sensors

1. Introduction

Estimating the frequency of a single-tone sinusoidal wave under noise is a fundamental problem in signal processing for communications, and its effect is extended to biomedical and power engineering [1,2].

Several classical methods were proposed to estimate the frequency of a single-tone sinusoid, mostly based on Fourier analysis [3].

Correlation methods can be very attractive for hardware implementation, IoT, sensors, and software-defined ratio (SDR), as they are much more computationally efficient than spectral techniques are. However, those techniques are far less accurate compared to Fourier-based techniques, especially under low values of signal-to-noise ratio (SNR) [4].

Phase-locked loops (PLLs) are widely used in communication systems to handle this problem. However, PLLs can be slower than spectral or correlative techniques as they need time for locking [5]. More samples are needed in digital PLLs (DPLLs) if more realizations are considered for better results, especially under a low SNR where noise significantly disturbs the locking process.

Recently, machine learning has emerged as a powerful tool for various tasks in many fields, including communications and signal processing [6–8]. Despite the increased

attention to machine learning, very few works considered its application to handle the problem of frequency estimation.

In [9], an approach based on deep learning for single-tone frequency estimation under noise was proposed. Unlike Fourier-based, PLL, or correlative techniques, the DL-based approach requires prior knowledge of the SNR. However, SNR estimation can be in error under varying noise conditions; therefore, many studies, including [9], assumed that SNR is constant during the period of frequency estimation.

In this work, we further investigate the performance DL for frequency estimation under insufficient SNR information. On the other hand, the effect of the number of nodes was also studied, and no significant change in performance was obtained after increasing the number of nodes in hidden layers. The contribution of this work has the following directions:

1. The performance of the DL approach is compared with the performance of still-active classical techniques that are based on Fourier analysis.
2. In many situations, SNR estimation can be inaccurate or unavailable. Hence, system performance is investigated in case of unavailable SNR information.
3. The DL approach is SNR-dependent; hence, an investigation of system performance under various SNRs is presented.
4. The more the nodes in the DL approach are, the better the accuracy of estimation could be. Hence, this work investigated the effect of changing the number of nodes in the hidden layers of the network.
5. The number of input samples (signal length or duration) has significant impact on the complexity and the performance of classical and DL-based methods. This point was fully investigated in this work.
6. The effect of different realizations while training was handled in the literature of DL-based approaches, as it is a necessary step in the training process. However, the possibility of different realizations in the working environment (application phase) was not previously handled. In this work, we discuss the effect of different realizations during the application phase.
7. The reduced complexity introduced by DL-based FE, in addition to avoiding complex-valued arithmetic operations, makes FE easier and cheaper for IoT communications, sensors, sensor networks, and software-defined radio (SDR). This work presents a discussion on such possibilities.

This paper is organized as follows. Section 2 presents related works; Section 3 presents the signal model and the problem definition; Section 4 reviews the most-effective classical techniques; and Section 5 addresses the yet-unhandled issues of the DL approach and presents the results in a comparative study that reveals the power of DL versus classical frequency-estimation techniques.

2. Motivation and Related Work

The problem of frequency estimation (FE) is often handled using classical Fourier and correlative techniques. However, some recent works handled this problem by neural networks and deep learning. Good accuracy was obtained; hence, the use of DL for frequency estimation is promising.

The authors in [9] designed a three-layer neural network to estimate the frequency of a sinusoid contaminated by a white-noise process at an SNR of 25 dB, where the trained model could estimate the frequency of any previously unseen noisy sinusoid in a short amount of time. However, this work did not address some important issues, including comparisons with classical techniques, system performance under wrong SNR estimation, system performance under various SNR values, and the effect of the number of nodes. Specifically, the following points are not clear in [9]:

1. The network was trained for a specific SNR. There should be a clarification whether different networks should be present in the case that different SNRs are expected.
2. The effect of SNR on estimation error was unclear.

3. The number of input nodes was chosen as N = 2000. It is not clear whether this choice of the input samples (nodes) is frequency-, duration-, or network-dependent.
4. The division of the frequency range was not clear. The effect of this division on estimation error should be addressed. The relation of this division to the time–frequency uncertainty principle should also be clarified.
5. The number of nodes in the hidden layers was 2. The effect of this number on estimation accuracy should be clarified. The possibility that this effect is SNR-dependent should be addressed.
6. Classical techniques for frequency estimation have been well-studied for decades. There should be a clear reasoning why one should choose DL-based estimation instead.

In [10], the authors presented an approach based on deep neural networks to estimate the fundamental frequency (F0), which is an essential acoustical feature to find the audible pitch level. This problem is classically dealt with using time-domain or correlative methods; however, the authors in [5] applied a different direction via deep learning where error was reduced as compared with classical techniques.

In [11], the authors presented an approach for estimating the frequency of linear frequency modulation (LFM) signals, a topic that has applications in radar and communication engineering. The approach utilized a convolutional neural network (CNN), while this problem was traditionally handled using time–frequency analysis [12]; however, this approach requires significant computational cost as it involves two-dimensional transforms.

3. Problem Statement

The single-tone sinusoidal generic model is given by

$$s(t) = a(t)\cos(2\pi f_o t + \theta_o), \tag{1}$$

where $a(t)$ is amplitude; normally, the maximal frequency in its spectrum is much lower than sinusoidal frequency f_o (i.e., slowly varying). θ_o is the initial phase.

As a carrier for information signals in communication systems via modulation, the sinusoidal carrier normally has constant amplitude, while the receiving device can apply amplitude estimation and a constant-gain multiplier to restore the original amplitude. Therefore, here, we consider sinusoidal model

$$s(t) = A\cos(2\pi f_o t + \theta_o) \tag{2}$$

with constant amplitude A.

The signal in Equation (2) is transmitted as a pilot to allow for the receiver to estimate the carrier frequency. However, such a single-tone signal can undergo frequency changes due to the Doppler effect and oscillator instabilities (see [1] and the references therein). Even minor change in carrier can cause demodulation problems. The performance of Orthogonal Frequency Division Multiplexing (OFDM) systems can deteriorate for improper carrier estimation [1]. Hence, accurate frequency estimation of the incoming carrier is very important for correct demodulation at the receiver.

However, communication channels add noise to transmitted signals, making frequency estimation a more challenging task. Therefore, the actual carrier model used in frequency estimation is as follows.

$$s(t) = A\cos(2\pi f_o t + \theta_o) + \eta(t), \tag{3}$$

where $\eta(t)$ represents the noise process. In most communication systems, noise is modelled as a Gaussian (normal) process with zero mean and variance that equals noise power σ^2 and the process is normally referred to as $\mathcal{N}(0, \sigma^2)$.

Signal-to-noise ratio (SNR) is defined as follows.

$$\text{SNR} \; = \; r \; = \; \frac{\text{signal power}}{\text{noise power}} \; = \; \frac{A^2/2}{\sigma^2} \tag{4}$$

and usually is expressed in decibels:

$$\text{SNR}_{\text{dB}} \; = \; 10 \; \log_{10}(\text{SNR}) \tag{5}$$

or simply

$$r_{\text{dB}} \; = \; 10 \; \log_{10}(r),$$

which is more convenient to logarithmically handle large and small values.

This work focuses on the problem of estimating frequency f_o of the single-tone model in Equation (3) under Gaussian noise $\eta(t)$. Our approach was to use a deep-learning (DL) neural network with multiple layers in this process, following recent works that confirmed the high accuracy of this approach [9–11]. Despite the pioneering role of these works, they did not address many fundamental issues in DL frequency estimation. We focus on clarifying these issues, and define the research problems as follows.

- Comparing the performance of DL-based approach with classical techniques.
- Investigating system performance in case of unavailable SNR information.
- Investigating system performance under various SNRs.
- Investigating the effect of changing the number of nodes in the hidden layers of the network.
- Investigating the effect of input signal length (duration) on the performance of both classical and DL-based methods. This factor has significant impact on overall performance and complexity.
- Investigating the effect of different realizations during the application phase (not only during the training phase).
- Investigating the impact of DL-based FE on IoT, sensors, sensor networks, and software-defined radio (SDR).

In addition to the above, investigating the effect of different realizations of the input signal in the application phase of the DL approach would be handled. It is known that different realizations of any statistical experiment would result in better estimation of its random variable. This point could be treated by increasing the number of input nodes; however, this direction would complicate the network structure and limit the choice of the user who may be satisfied with a user-defined accuracy level.

The next section presents an overview of main well-known approaches in classical FE for single-tone signals.

4. A Brief Overview of Classical FE for Single Tones

Literature surveys of classical frequency-estimation methods are presented in [2,13,14]. In [1], an iterative approach for single-tone frequency estimation was presented; however, despite their benefits in accuracy, such approaches can be computationally expensive or complex.

In [15–17] and the references therein, good DFT interpolators were presented; however, despite some accuracy gains, they were computationally more expensive than earlier methods in the literature were.

Reduced complexity is a property that can be very useful for some resource-limited systems such as wireless sensor networks [18]. The complexity–accuracy–speed trade-off is the selection criterion in this work.

Here, a brief overview of the most accurate, computationally efficient classical approaches is presented. All accurate classical FE methods are based on using the discrete Fourier transform (DFT) of the signal:

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-j2\pi kn/N} \tag{6}$$

where N is the number of input samples. Various methods were designed for FE on the basis of interpolating the DFT of the sinusoid. The most accurate and computationally efficient methods under additive Gaussian noise are as follows.

4.1. Maximum of DFT Estimator

This method is based on the location (index) of the maximum of the absolute of the DFT, given by

$$I_m = \arg\left\{ \max_k [|S(k)|] \right\} \tag{7}$$

since the sinusoid gives a delta spectrum. Estimated frequency is given by

$$F_o = \frac{I_m \cdot f_s}{N}, \tag{8}$$

where f_s is sampling frequency.

The error of estimation at SNR = r (dB) is defined in this paper as

$$e(r) = \frac{|F_o - f_o|}{f_o}, \tag{9}$$

and the above definition of the error was used to evaluate the performance of DL approach versus existing methods. Another performance criterion is the computational complexity of the method.

4.2. Quadratic Interpolator

This approach designs a quadratic interpolation for Fourier transform (DFT) using three points near the index of the maximum of |DFT|, i.e., the following points in the DFT domain:

$$(I_m - 1, S_1 = |S_{I_m-1}|), (I_m, S_2 = |S_{I_m}|), (I_m + 1, S_3 = |S_{I_m+1}|),$$

where I_m is given in Equation (7). The estimated frequency is given by

$$F_o = \frac{I_o \cdot f_s}{N}, \tag{10}$$

where $I_o = I_m + \delta; \delta = (S_3 - S_1) / [2 * (2 * S_2 - S_1 - S_3)]$.

4.3. Barycentric Estimator

This estimator is similar to the quadratic estimator, with the estimated frequency given by

$$F_o = \frac{I_b \cdot f_s}{N}, \tag{11}$$

where $I_b = I_m + \alpha; \alpha = (S_3 - S_1) / [S_1 + S_2 + S_3]$.

During this work, the barycentric estimator gave the best results in terms of error among the selected classical techniques.

The next section presents the structure and training details of the deep-learning network that was adopted for the FE of single-tone signals.

5. Deep Learning for Single-Tone FE: Network Structure and Training

In this section, the structure of the DL approach for the frequency estimation of single tones is analyzed and justified. In the DL-based approach, complexity only exists during the training phase and not in the actual network structure. The significant reduction in complexity can make the DL approach for FE suitable for IoT communications involving wireless sensor networks (WSNs).

5.1. DL Network Structure

A deep neural network with two hidden layers is shown in Figure 1, where arrows represent the multiplication of relevant items by specific weights and adding biases, using the MATLAB tansig (or tanh) function as the pointwise activation function in hidden layers as follows.

$$\mathbf{\Phi}\{Z_{\mathfrak{N}\times 1}\} \;=\; \mathbf{\Phi}\{[z_i|i \;=\; 1:\mathfrak{N}]\} \;=\; [\phi(z_i)]_{\mathfrak{N}\times 1} \;=\; \left[\frac{2}{1+e^{-2z_i}}-1\right]_{\mathfrak{N}\times 1}, \tag{12}$$

where $\mathfrak{N}$ is the number of layer output nodes.
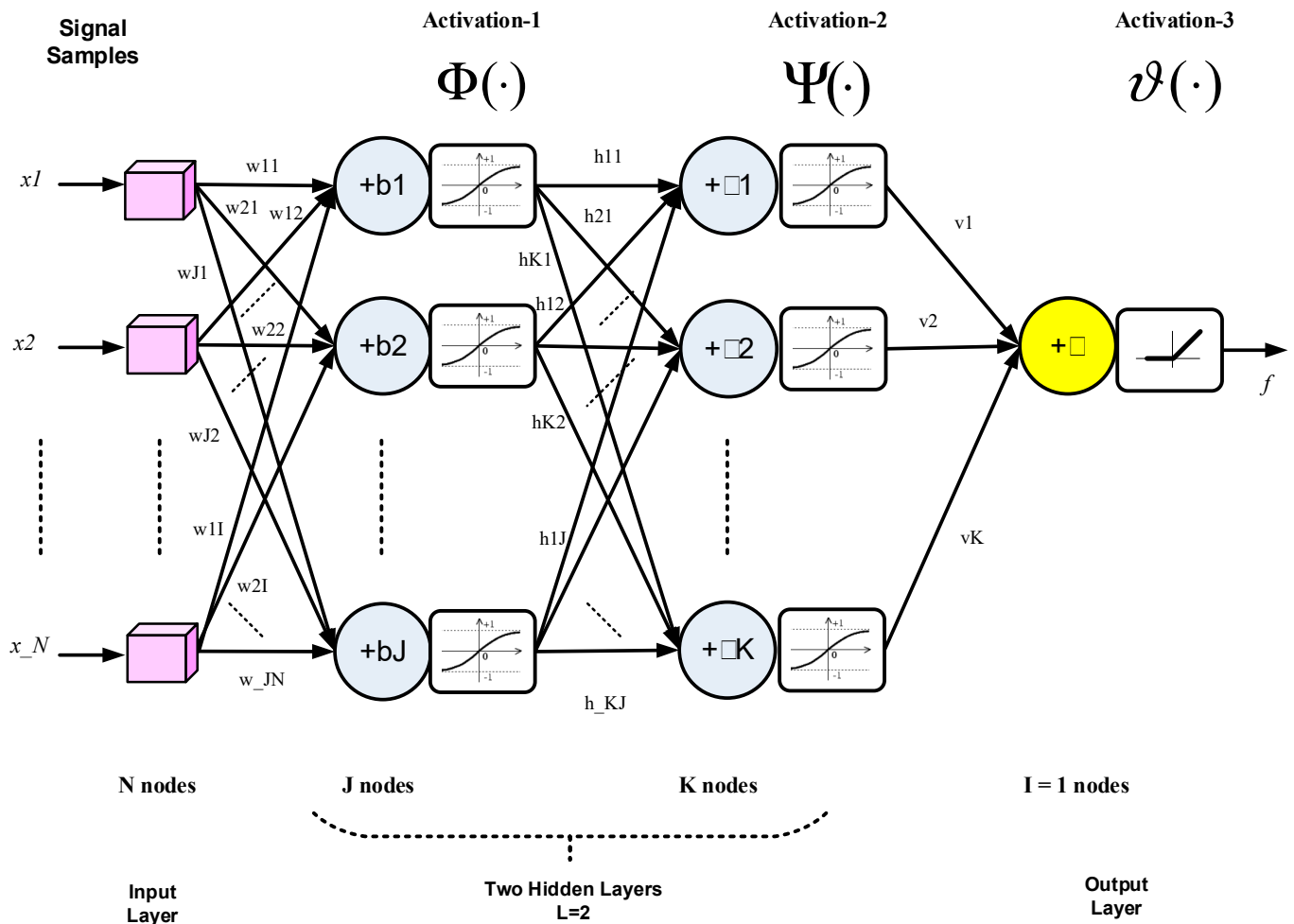

Figure 1. Structure of two-layer network for deep-learning (DL)-based frequency estimation (FE).

Tanh activation for hidden layers is used to prevent results from expanding in magnitude and normalizing them in the $[-1, 1]$ range. Both positive and negative values are needed for the proper adjustment of the network, unlike the case in the output layer, where

a single output (a decision) is obtained using the positive linear activation function (ReLU) to give a proportional value to the input frequency as follows.

$$\vartheta(x) \;=\; \max(0, x) \;=\; \left\{ \begin{array}{ccc} x & ; & x \geq 0 \\ 0 & ; & x < 0 \end{array} \right\}, \tag{13}$$

where ReLU is used because the actual value of the frequency is needed, knowing that the frequency is always positive.

The operation of the 2-layer network in Figure 1 can be expressed as follows.

$$f \;=\; \vartheta\big[\boldsymbol{V}_{1\times K}\cdot\boldsymbol{\Psi}\big\{\boldsymbol{H}_{K\times J}\cdot\boldsymbol{\Phi}\big(\boldsymbol{W}_{J\times N}\cdot\boldsymbol{X}_{N\times 1} + \mathbf{b}_{J\times 1}\big) + \boldsymbol{\beta}_{K\times 1}\big\} + \lfloor_{1\times 1}\big], \tag{14}$$

where X is the input vector; W, H, and V are the weight matrices of the first hidden, second hidden, and output layers, respectively; N, J, K, are the numbers of nodes of the input, first hidden, and second hidden and layers, respectively; and $\mathbf{b}$, $\boldsymbol{\beta}$, and $\lfloor$ are the biases of the first hidden, second hidden, and output layers, respectively. The number of nodes in the output layer is $I = 1$ as a scalar is needed to estimate the input frequency. Bold type was used for matrices and vector functions, and plain-text symbols for scalar functions and variables.

The activation functions handle the relevant vectors pointwise as shown in Figure 1, while output variable f is a scalar that represents the input frequency.

The minimal number of hidden layers L for the successful extraction of input features is $L = 2$. A larger L may slightly improve performance, but it increases complexity.

5.2. Training Data

Training data were samples that were simulated using the model in Equation (3) on MATLAB (under Academic License 40635944). The number and magnitude of these samples are dependent on signal duration T, sampling frequency f_s, and the signal-to-noise ratio (SNR).

Data samples for each T and SNR were divided into three groups: 70% for training, 15% for testing, and 15% for validation.

The training function of the DL network was chosen to be the MATLAB trainscg function, which updates weight and bias values according to the scaled conjugate gradient method, with learning rate of $L_r = 0.1$ and number of epochs $p = 100$.

As changes in carrier frequency are expected to be relatively small (mostly due to the Doppler effect [19]), a small low-frequency Hz range [20–22] was considered in this experiment, with a sampling frequency of $T_s = 0.01$ Hz. This makes normalized frequency f/f_s be in the range of [0.23, 0.25]. This 9is the normalized frequency that is considered while handling a frequency-estimation problem [20]. On the other hand, small ranges of small frequencies are expected in power systems (e.g., 50 Hz) and biosystems (for example, electroencephalograms (EEG), in the range of 0–40 Hz) [21,22].

Higher ranges are possible but require more training samples N and thereby more complex networks; for example, handling a maximal frequency of 100kHz may require $N = 2000$ input nodes [9].

The necessary number of input samples that affects DL performance is related to number γ of complete periods (cycles) of the input sinusoid that are included in the training process, which can be expressed as

$$\gamma \;=\; \frac{T}{T_o} \;=\; \frac{T/T_s}{T_o/T_s} \;=\; \frac{N}{\nu_o}, \tag{15}$$

where ν_o is the number of samples in one cycle of the input signal with frequency f_o and period T_o, given by

$$\nu_o \;=\; \frac{T_o}{T_s} \;=\; \frac{f_s}{f_o}, \tag{16}$$

from which the number of cycles used in [9], which used $T_s = 1$ μs (hence, $f_s = 1 \times 10^6$ Hz) and a maximal input frequency of $f_o = 1 \times 10^5$ Hz, can be calculated as

$$\gamma = \frac{N}{\nu_o} = N\frac{f_o}{f_s} = 2000 \frac{1 \times 10^5}{1 \times 10^6} = 200.$$

In this work, the number of input cycles used to train the DL reached the limit of

$$\gamma = N\frac{f_o}{f_s} = 11\frac{25}{100} = 2.75,$$

giving reasonable results with almost 3 input cycles, as shown in Section 6.

Hence, the use of N = 2000 in [9] is an overestimation of the number of input samples that are necessary to train the DL network. While this overestimation may give the advantage of minor improvement in error performance, it is at the expense of increasing network complexity, which may not be attractive for resource-limited systems such as sensors in a wireless sensor network (WSN).

Generally, the choice of the proper number of input samples should be based on the value of γ, and this process represents a trade-off between complexity and accuracy.

The selected frequency range was divided with a frequency step of $df = \delta/\mu$, with $\delta = 25 - 23 = 2$ and $\mu = 40$ in the simulations.

Of course, more accuracy can be obtained with a larger μ, but this can be at the expense of more nodes in the network.

This frequency step is time-independent; hence, it is a design problem, unlike frequency step $\Delta f = 1/T$ in DFT-based techniques, which is time-dependent and follows the Heisenberg uncertainty principle in the time–frequency domain [23].

For each incremental frequency $\{f_i = 23 + i \cdot df | i = 0 : \mu\}$ in the selected frequency range, a number of realizations $R = 100$ is generated, so that the training vector is

$$X_t = \begin{bmatrix} S_0 & S_2 & \cdots & S_\mu \end{bmatrix}_{N \times (\mu+1)R}; \ S_i = \begin{bmatrix} s_{i1} & s_{i2} & \cdots & s_{iR} \end{bmatrix}_{N \times R} \tag{17}$$

with labels

$$L_t = \begin{bmatrix} L_0 & L_2 & \cdots & L_\mu \end{bmatrix}_{1 \times (\mu+1)R}; \ L_i = i * \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}_{1 \times R}, \tag{18}$$

where s_{ik} is the $N \times 1$ column that represents the kth realization of Equation (3) with a frequency f_i under the same SNR. Hence, there is a total of $\mu + 1$ classes. A larger number of realizations R during the training phase implies better training.

5.3. Training Function: Scaled Conjugate Gradient Backpropagation

The training function is the most important part of network design as it is responsible for finding the proper weight and bias values. Several training functions were tried for the FE problem in this work, but most either failed or gave a large error (insufficient performance). The best function for FE problems is the scaled conjugate gradient (SCG) backpropagation algorithm [24,25].

SCG's superior performance is because it user-independently updates its parameters at each iteration [24]. In addition, SCG uses multiple adaptivity where the step size and search direction are updated as shown in Equations (29) and (30). Several applications confirmed this (see, for example, [26]).

The SCG optimization function combines the Levenberg–Marquardt (LM) algorithm (which is, in turn, a combination of gradient descent and the classical Gauss–Newton algorithm) with the conjugate-gradient approach [24]. The training process utilizes iterative algorithms to update the weights and biases of the network based on multivariate error function

$$e(v) = \frac{1}{2} \sum_{k=1}^{R} e_k^2 = \frac{1}{2} \sum_{k=1}^{R} (f_k - f_o)^2, \tag{19}$$

where f_k is the output of the network at the *k*th realization. All training algorithms minimize this global error by the optimal adjustment of multivariate

$$v = [w\,b]_{1\times\xi},$$ (20)

which includes weights (*w*) and biases (*b*) with $\xi/2$ = number of nodes.

The update rule of the LM algorithm is given at the *i*th iteration (epoch) by

$$v_{i+1} = v_i - \left[\nabla^2 e(v_i) + L_r\mathbb{I}\right]^{-1}\nabla e(v_i),$$ (21)

where ∇e is the gradient operator, damping parameter L_r controls the update process (the learning rate), and $\mathbb{I}$ is the identity matrix. The Hessian matrix is approximated by

$$\nabla^2 e(v_i) \approx \mathbb{J}^T\mathbb{J},$$ (22)

while the gradient can be written as

$$g_i = \nabla e(v_i) = \mathbb{J}^T e(v_i),$$ (23)

where

$$\mathbb{J} = \left[\frac{\partial e_k}{\partial v_j}\right]_{R\times\xi}$$ (24)

is the Jacobian matrix. Hence, LM weight-bias update can be written as follows.

$$v_{i+1} = v_i - \left[\mathbb{J}^T\mathbb{J} + L_r\mathbb{I}\right]^{-1}\mathbb{J}^T e(v_i).$$ (25)

The optimal solution is *v*, which makes $g = \nabla e(v) = 0$, and v_o is random.

Learning rate L_r balances the performance between Newton's method (when it is zero) and the gradient descent (when it is large).

In the first iteration (epoch) of the SCG, v_o is chosen at random. Then, LM is used to find the steepest descent, which is the negative of LM gradient

$$p_o = -g_o.$$ (26)

Then, optimal step α_k [26] is computed as follows.

$$\alpha_k = -\frac{v_k^T\rho_k}{v_k^T G_k},$$ (27)

$$G_k = \lim_{\epsilon\to 0}\frac{g_k(v_k + \epsilon\cdot v) - g_k(v_k - \epsilon\cdot v)}{2\epsilon},$$ (28)

with *v* = arbitrary ξ-dimensional vector and ϵ = a small positive number. The two gradients $g_k(v_k + \epsilon\cdot v)$ and $g_k(v_k - \epsilon\cdot v)$ are evaluated by LM. The new weight-bias vector is given by

$$v_{k+1} = v_k + \alpha_k\rho_k.$$ (29)

Then, gradient g_{k+1} is evaluated by LM at v_{k+1}.

Now, a new search direction is conjugate to the previous one using the following relation.

$$p_{k+1} = -g_{k+1} + \beta_k p_k,$$ (30)

where parameter β_k is given by

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k},$$ (31)

which is known as the Fletcher–Reeves formula [27]. Then, the update continues by going back to Equation (27) to find new step α_{k+1}.

6. Results: Deep Learning vs. Classical Single-Tone FE

In this section, we analyze the performance of the DL approach for the frequency estimation of single tones in comparison with computationally efficient classical techniques.

For each SNR, a different network is trained to recognize frequencies in the specified frequency range. As the SNR may change in communication systems, this result may imply that a complicated set of DL-based estimators is necessary for proper functioning, and this complexity is not necessary.

The number of signal samples N used in the design of the estimation system has significant impact on FE performance in both DFT- and DL-based techniques. Therefore, a larger signal duration T (allowed for by the estimation system design) gives a larger $N = T/T_s$, and thereby better estimation.

Even better results can be obtained via more realizations. During the training phase, a larger number of realizations r implies better training. In the testing phase (actual working environment), obtaining a number ρ of different realizations is possible via dividing the input samples into sets of specific length that equal signal length N, which was selected during training (i.e., the number of input nodes to the trained DL network). Hence, for better frequency estimation, the transmitter should initially send the carrier single tone for sufficient sending duration P that includes a multiple ρ of training duration T (of which the sample length is $N = T/T_s$), i.e., $P = \rho T$ or $\rho N = P/T_s$. The final frequency estimate is the average of estimates over ρ realizations.

6.1. Performance under Different Input Lengths

First, we compared the performance of the DL-based FE with the selected classical DFT-based techniques for different values of the training input size (N). Figure 2 shows the performance of DL method with $J = K = 5$ versus selected classical techniques. The sinusoidal frequency was selected to be $f_o = 24.41$ Hz. For a small N, the classical DFT-based approach failed to correctly estimate the input frequency.
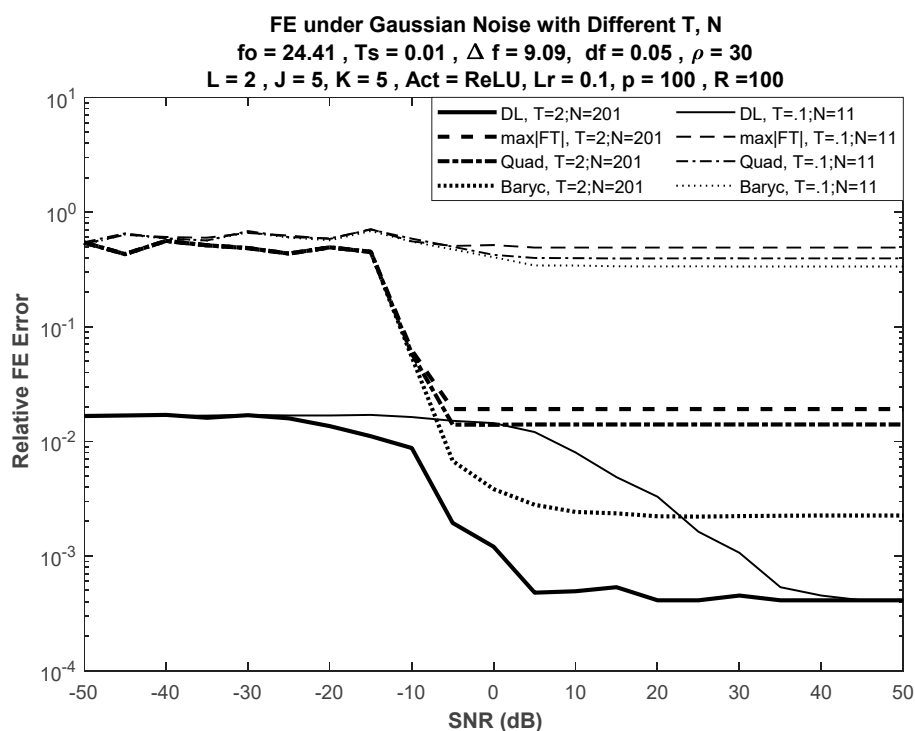

Figure 2. Performance of FE methods for different input lengths. Bold curves are for larger duration $T = 2$ s, $N = 201$ input samples; plain curves are for smaller duration $T = 0.1$ s, $N = 11$.

The number of realizations during the training phase was selected to be $R = 100$. Considering different realizations in the training phase improves the network design, so that it would be ready to handle different situations in the application phase. During the application (working) phase, the number of realizations was selected to be $\rho = 30$. These working realizations improve estimation performance. In this paper, network performance was considered for $\rho = 30$. This choice of ρ is user-dependent and has no effect on network complexity.

Figure 3 shows the performance of the DL-based approach versus the best of the selected DFT-based approaches, the barycentric approach, for different input lengths, while Figure 4 shows the performance of the DL-based approach versus the barycentric approach with fewer nodes in the network, such that $J = K = 3$.
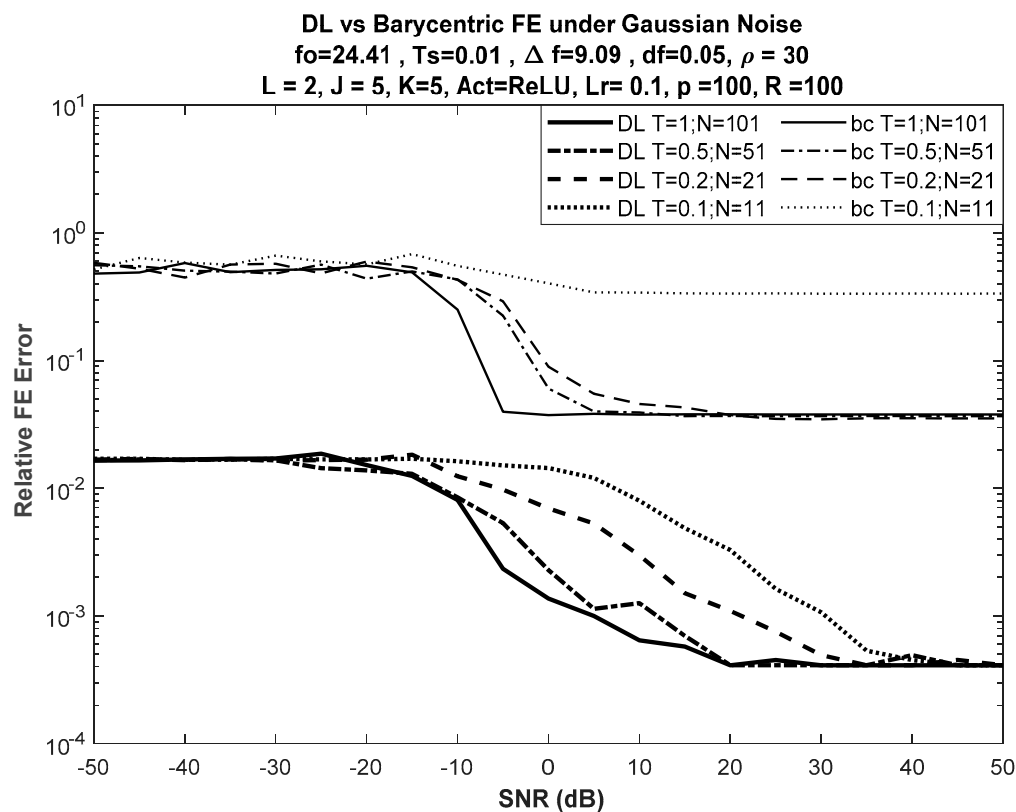

Figure 3. DL vs. barycentric discrete Fourier transform (DFT)-based FE for different input lengths with $J = K = 5$. Note that DFT-based FE failed when $N = 11$, while DL-based FE gave reasonable results even at low signal-to-noise ratio (SNR).

No significant difference was seen in error performance, as shown in Figure 4 (for $J = K = 3$), when compared with the error performance shown in Figure 3 (for $J = K = 5$). Despite the reduction in the number of nodes in both hidden layers of the network from 5 to 3, the error curves had almost the same trajectories versus SNR.
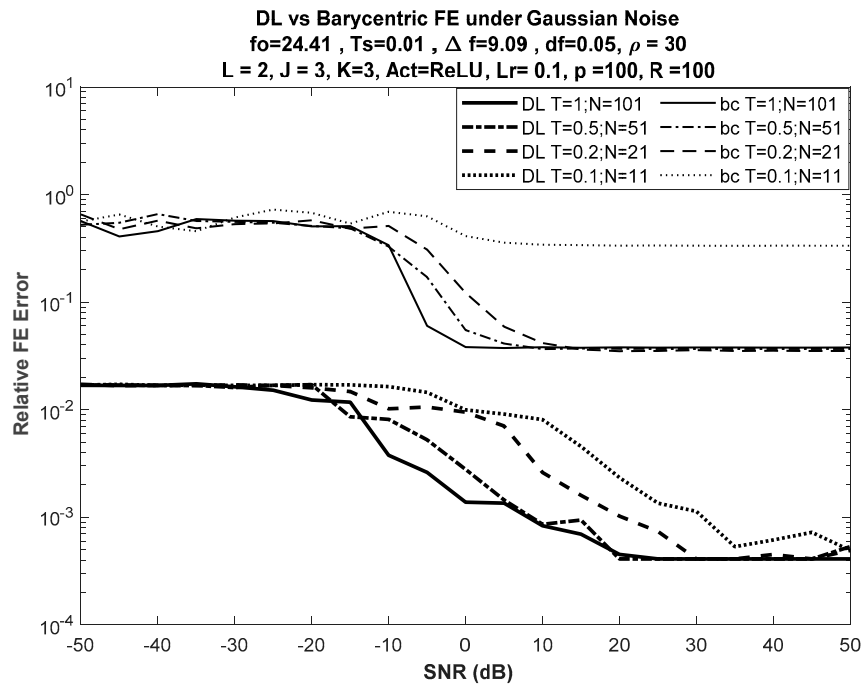
DL vs Barycentric FE under Gaussian Noise
fo=24.41 , Ts=0.01 , Δ f=9.09 , df=0.05, ρ = 30
L = 2, J = 3, K=3, Act=ReLU, Lr= 0.1, p =100, R =100

Figure 4. DL vs. barycentric DFT-based FE for different input lengths with $J = K = 3$. Bold curves used for DL.

6.2. Performance under Insufficient SNR Information

SNR estimation is necessary for many purposes in communication systems, and it is needed for DL frequency estimation, as training is performed under a specific SNR. However, in some systems (e.g., some sensor networks), limited computational capabilities may not allow for SNR estimation except probably for a rough estimate. Knowing the exact value of SNR did not cause the performance of DL estimation to deteriorate, where only minor changes in the performance were noticed, as shown in Figure 5. Hence, the same network could be used for FE under different SNRs.
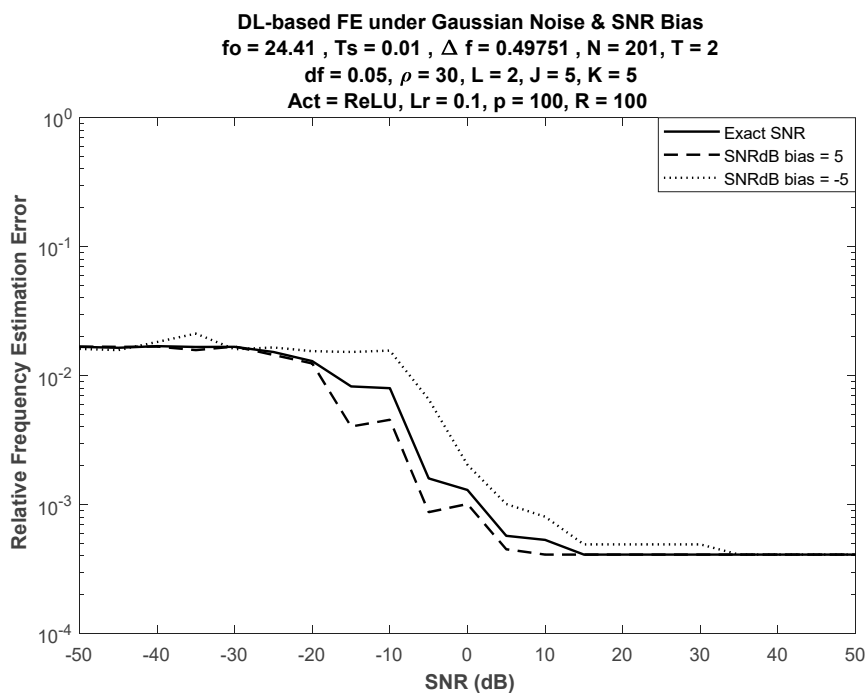
DL-based FE under Gaussian Noise & SNR Bias
fo = 24.41 , Ts = 0.01 , Δ f = 0.49751 , N = 201, T = 2
df = 0.05, ρ = 30, L = 2, J = 5, K = 5
**Act = ReLU, Lr = 0.1, p = 100, R = 100**

Figure 5. DL-based FE under bias from training SNR with $J = K = 5$.

Relative deviation ν of DL performance for FE, averaged over M values of SNR, is defined as

$$\nu = \frac{1}{M}\sum_{k=1}^{M}\frac{e_b(k)}{e_o(k)}, \tag{32}$$

where $e_o(k)$ and $e_b(k)$ are the average errors at the kth SNR value (dB) under normal operation (correct SNR information) and biased SNR, respectively. Figure 6 shows relative deviation when $J = K = 5$. Nonsymmetric deviation was expected for positive and negative bias due to the nonsymmetry of the FE error performance curves (see Figures 2–4).

Performance of DL-based FE under Gaussian Noise & SNR Bias
fo = 24.41 , Ts = 0.01 , $\triangle$ f = 0.49751 , N = 201, T = 2
df = 0.05, ρ = 30, L = 2, J = 5, K = 5
Act = ReLU, Lr = 0.1, p = 100, R = 100
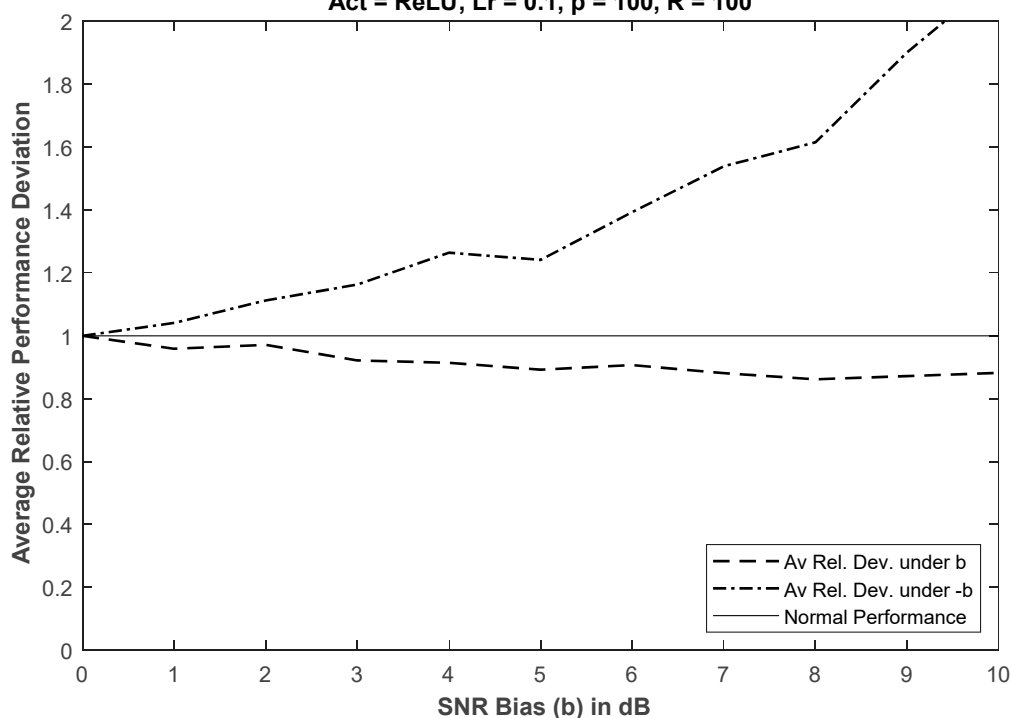
Figure 6. Average performance deviation of DL-based FE versus SNR bias.

6.3. Effect of Increasing Nodes at Hidden Layers

Generally, no significant performance difference was found when increasing the number of nodes in the hidden layers, as shown in Figure 7, where the performance comparison versus the barycentric method is presented using DLs with $J = K = 5$ and $J = K = 3$.
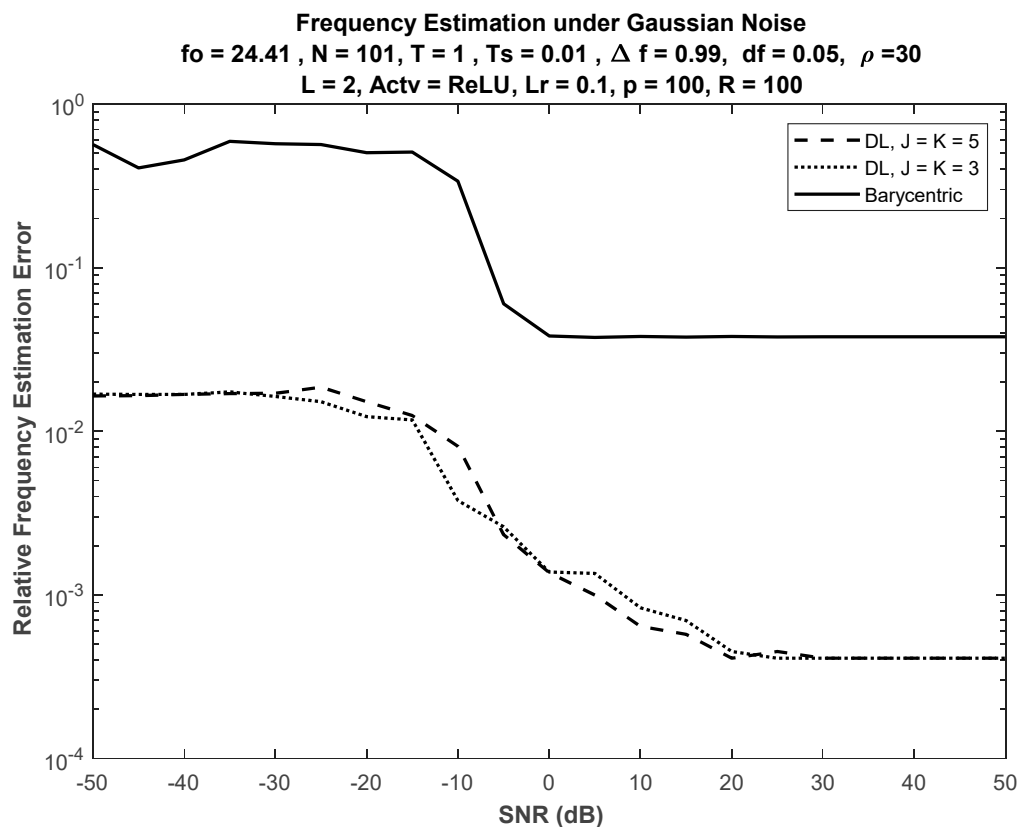
Figure 7. Performance of DL-based FE under different sizes of hidden layers. No significant difference in DL performance between networks with sizes $J = K = 5$ and $J = K = 3$.

6.4. Impact on IoT, Sensors, and SDR

Reduced complexity can lead to the efficient implementation of FE, since DL does not need general-purpose digital multipliers if implemented as a multibit system, where trained weights can be implemented using look-up tables.

On the other hand, cheap and hardware-efficient implementation using short word length (SWL) is possible due to reduced complexity, where complicated digital multipliers are replaced by simple multiplexers.

Another hardware gain is because DFT techniques require the use of a complex-valued arithmetic, while DL systems use only a real-valued arithmetic. The implementation of a large DFT is a very complicated task for both hardware and software designs despite the significant efficiency provided via the latest versions of the fast Fourier transform (FFT) algorithm, which reduces the number of (complex) multiplications from $\mathcal{O}\{N^2\}$ to $\mathcal{O}\{N \log_2(N)\}$ [28].

From this work and [9–11], it can be inferred that N is proportional to input frequency f_o; hence, for high frequencies, size N can be large if DL is used, and much larger for classical FE if the same accuracy as that of DL is required, calling for very complicated DFT implementation in the complex-valued domain.

FFT implementation via parallel-computing algorithms can alleviate the time cost [20]; however, this approach is very expensive, power-consuming, and hardware-costly, which may not be suitable for many systems, especially in emerging IoT sensor applications.

Large-scale sensor applications require tiny, low-power, low-cost implementation, such as sensors used for healthcare monitoring, the early detection and monitoring of forest fires, natural-disaster detection, and large-scale surveillance for security applications.

Complicated, complex-valued operations increase both system size and complexity, and power consumption, which is a crucial factor in sensors deployed in WSNs, as the lifetime of these sensors is limited by the irreplaceable battery life [18].

For DL-based approaches, the number of (real) multiplications is mostly at the input stage, which is N, while other node multiplications are normally as low as 6 real-valued multiplications (which are trivial operations as they use look-up tables instead of digital multipliers); hence, the overall number of real-valued multiplications is $\mathcal{O}\{N\}$.

Whether implemented using SWL or multibit technologies, this reduced complexity (versus the complicated, complex-valued classical FE techniques) makes the DL-based approach the most suitable FE technology in sensors and sensor networks, which are inherently linked to modern IoT applications.

Similar gains are obtained if implementation is performed via software, where reduced complexity leads to significant time gain, a factor that is attractive for software-defined radio (SDR) communications.

6.5. Future Directions: DL in the SWL Domain

The authors are currently investigating the possible implementation of the DL-based FE using short word length (SWL) where single-bit and ternary technologies are considered.

As the training phase can be performed offline, training could be accomplished using multibit computer simulation; however, training data should be converted into the SWL format. In this case, the new factor of quantization noise emerges, and this can be handled by better noise shaping via the proper selection of oversampling ratio (OSR), the quantization process, and the filter design of the sigma–delta modulator (SDM).

As adaptivity is now performed offline, this DL design for SWL could be even less complicated than the design of adaptivity in SWL systems is, first introduced in [29] as a result of SWL research supported by the Australian Research Council via discovery grant DP0557429.

The successful SWL implementation of DL-based FE could transform many other resource-limited IoT and sensor applications after the advent of DL techniques into the SWL domain.

7. Conclusions

This paper presented comprehensive analysis of the performance of a deep-learning based approach for the frequency estimation of a single-tone sinusoid. This work fills in the gaps of unhandled issues in recently proposed methods based on deep learning. The paper presented analysis of the frequency-estimation error under a range of signal-to-noise ratios (SNRs). Hidden layers can be as few as two, each with two or three nodes. The number of input samples (nodes) can be very small as compared to that in DFT-based classical methods, with much better error performance. In addition, performance was not significantly degraded in the case of wrong information about the actual value of the SNR. These results can reduce the complexity, power consumption, and cost of the communication system, properties that can be very useful for systems with limited memory and computational capabilities, such as wireless sensor networks (WSNs), which are inherently linked to current and future applications on the Internet of Things. Reduced complexity without using a complex-valued arithmetic also makes the DL technology for frequency estimation suitable for software-defined radio. Exploiting this reduced complexity to introduce DL in the short-word-length (SWL) domain was presented as a future direction.

References

1. Serbes, A. Fast and Efficient Sinusoidal Frequency Estimation by Using the DFT Coefficients. *IEEE Trans. Commun.* **2018**, *67*, 2333–2342. [CrossRef]
2. Boashash, B. (Ed.) *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*; Elsevier: Amsterdam, The Netherlands, 2016.
3. Brown, R.D., III; Liao, Y.; Fox, N. Low-Complexity Real-Time Single-Tone Phase and Frequency Estimation. *IEEE Mil. Commun.* **2010**.
4. Yang, C.; Wei, G.; Chen, F.-J. An Estimation-Range Extended Autocorrelation-Based Frequency Estimator. *EURASIP J. Adv. Signal Process.* **2009**, *961938*, 1–7. [CrossRef]
5. Al-Araji, S.; Hussain, Z.M.; Al-Qutayri, M.A. *Digital Phase Lock Loops: Architectures and Applications*; Springer: Berlin/Heidelberg, Germany, 2006.
6. Cao, Y.; Geddes, T.A.; Yang, J.Y.H.; Yang, P. Ensemble deep learning in bioinformatics. *Nat. Mach. Intell.* **2020**, *2*, 500–508. [CrossRef]
7. Kour, H.; Gondhi, N. Machine Learning Techniques: A Survey. In *Innovative Data Communication Technologies and Application, ICIDCA 2019, Lecture Notes on Data Engineering and Communications Technologies*; Raj, J., Bashar, A., Ramson, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 46.
8. Boutaba, R.; Salahuddin, M.A.; Limam, N.; Ayoubi, S.; Shahriar, N.; Estrada-Solano, F.; Caicedo, O.M. A Comprehensive Survey on Machine Learning for Networking: Evolution, Applications and Research Opportunities. *J. Internet Serv. Appl.* **2018**, *9*, 1–99. [CrossRef]
9. Sajedian, I.; Rho, J. Accurate and Instant Frequency Estimation from Noisy Sinusoidal Waves by Deep Learning. *Nano Converg.* **2019**, *6*, 1–5. [CrossRef] [PubMed]
10. Xu, S.; Shimodaira, H. Direct F0 Estimation with Neural-Network-Based Regression. In Proceedings of the 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019), Graz, Austria, 15–19 September 2019.
11. Chen, X.; Jiang, Q.; Su, N.; Chen, B.; Guan, J. LFM Signal Detection and Estimation Based on Deep Convolutional Neural Network. In Proceedings of the APSIPA Annual Summit and Conference, Lanzhou, China, 18–21 November 2019.
12. Hussain, Z.M.; Boashash, B. The T-class of time–frequency distributions: Time-only kernels with amplitude estimation. *J. Frankl. Inst.* **2006**, *343*, 661–675. [CrossRef]
13. Boashash, B. Estimating and Interpreting the Instantaneous Frequency of a Signal-Part 1: Fundamentals. *Proc. IEEE* **1992**, *80*, 520–538. [CrossRef]
14. Almoosawy, A.N.; Hussain, Z.M.; Murad, F.A. Frequency Estimation of Single-Tone Sinusoids under Additive and Phase Noise. *Int. J. Adv. Comput. Sci. Appl.* **2014**.
15. Bai, G.; Cheng, Y.; Tang, W.; Li, S.; Lu, X. Accurate Frequency Estimation of a Real Sinusoid by Three New Interpolators. *IEEE Access* **2019**, *7*, 91696–91702. [CrossRef]
16. Fan, L.; Qi, G.; Xing, J.; Jin, J.; Liu, J.; Wang, Z. Accurate Frequency Estimator of Sinusoid Based on Interpolation of FFT and DTFT. *IEEE Access* **2020**, *8*, 44373–44380. [CrossRef]
17. Çagatay, C. A Method for Fine Resolution Frequency Estimation from Three DFT Samples. *IEEE Signal Process. Lett.* **2011**, *18*, 351–354. [CrossRef]
18. Hussain, Z.M. Energy-Efficient Systems for Smart Sensor Communications. In Proceedings of the IEEE 30th International Telecommunication Networks and Applications Conference (ITNAC2020), Melbourne, Australia, 25–27 November 2020.
19. Rice, F.; Cowley, B.; Moran, B.; Rice, M. Cramer-Rao lower bounds for QAM phase and frequency estimation. *IEEE Trans. Commun.* **2001**, *49*, 1582–1591. [CrossRef]
20. So, H.; Chan, F.K.; Sun, W. Efficient frequency estimation of a single real tone based on principal singular value decomposition. *Digit. Signal Process.* **2012**, *22*, 1005–1009. [CrossRef]
21. Escobar, J.J.M.; Matamoros, O.M.; Padilla, R.T.; Hernández, L.C.; Durán, J.P.F.P.; Martínez, A.K.P.; Reyes, I.L.; Espinosa, H.Q. Biomedical Signal Acquisition Using Sensors under the Paradigm of Parallel Computing. *Sensors* **2020**, *20*, 6991. [CrossRef] [PubMed]
22. Xia, Y.; He, Y.; Wang, K.; Pei, W.; Blazic, Z.; Mandic, D.P. A Complex Least Squares Enhanced Smart DFT Technique for Power System Frequency Estimation. *IEEE Trans. Power Deliv.* **2017**, *32*, 1270–1278. [CrossRef]
23. Cohen, L. *Time-Frequency Analysis*; Prentice Hall: Hoboken, NJ, USA, 1995.
24. Moller, M.F. A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Netw.* **1993**, *6*, 525–533. [CrossRef]

25.	The MathWorks. Trainscg: Scaled Conjugate Gradient Backpropagation, Available from MATLAB Documentation, Also at Math-Works Website. Available online: https://nl.mathworks.com/help/deeplearning/ref/trainscg.html (accessed on 2 April 2021).

26.	Vanus, J.; Nedoma, J.; Fajkus, M.; Martinek, R. Design of a New Method for Detection of Occupancy in the Smart Home Using an FBG Sensor. *Sensors* **2020**, *20*, 398. [CrossRef]

27.	Fletcher, R.; Reeves, C.M. Function Minimization by Conjugate Gradients. *Comput. J.* **1964**, *7*, 149–154. [CrossRef]

28.	Takahashi, D. *Fast Fourier Transform Algorithms for Parallel Computers*; Springer: Berlin/Heidelberg, Germany, 2019.

29.	Sadik, A.; Hussain, Z.; O'Shea, P. Adaptive algorithm for ternary filtering. *IET Electron. Lett.* **2006**, *42*, 420. [CrossRef]