Rapid parameter estimation of discrete decaying signals using autoencoder networks

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 045024 https://doi.org/10.1088/2632-2153/ac1eea

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
29 June 2021

REVISED
3 August 2021

ACCEPTED FOR PUBLICATION
18 August 2021

PUBLISHED
21 September 2021

PAPER

Rapid parameter estimation of discrete decaying signals using autoencoder networks

Jim C Visschers[1,2,*], **Dmitry Budker**[1,2,3] **and Lykourgos Bougas**[1,2,*]

[1] Institut für Physik, Johannes Gutenberg Universität-Mainz, 55128 Mainz, Germany
[2] Helmholtz-Institut Mainz, GSI Helmholtzzentrum für Schwerionenforschung GmbH, Mainz 55128, Germany
[3] Department of Physics, University of California, Berkeley, CA 94720-300, United States of America
[*] Authors to whom any correspondence should be addressed.

E-mail: jvisschers@uni-mainz.de **and** lybougas@uni-mainz.de

Keywords: signal processing, data analysis, statistics and probability, machine learning, parameter estimation

Abstract

In this work we demonstrate the use of neural networks for rapid extraction of signal parameters of discretely sampled signals. In particular, we use dense autoencoder networks to extract the parameters of interest from exponentially decaying signals and decaying oscillations. By using a three-stage training method and careful choice of the neural network size, we are able to retrieve the relevant signal parameters directly from the latent space of the autoencoder network at significantly improved rates compared to traditional algorithmic signal-analysis approaches. We show that the achievable precision and accuracy of this method of analysis is similar to conventional algorithm-based signal analysis methods, by demonstrating that the extracted signal parameters are approaching their fundamental parameter estimation limit as provided by the Cramér–Rao bound. Furthermore, we demonstrate that autoencoder networks are able to achieve signal analysis, and, hence, parameter extraction, at rates of 75 kHz, orders-of-magnitude faster than conventional techniques with similar precision. Finally, our exploration of the limitations of our approach in different computational systems suggests that analysis rates of $>$200 kHz are feasible using neural networks in systems where the transfer time between the data-acquisition system and data-analysis modules can be kept below $\sim$3 μs.

1. Introduction

Machine learning (ML) is becoming a widespread method for the generation, and analysis of big data. ML uses networks of interconnected neurons (i.e. neural networks) that, much like real brains, recognize patterns in data structures [1, 2]. Generally, these neural networks are first trained in situations where the desired action/output of the network is known and subsequently used in similar, real-world situations. Research fields where a precise mathematical description of the problem is difficult—if not impossible—stand much to gain from implementing ML solutions. Areas such as image processing [3, 4] or text generation and analysis [5] have seen innovations that would not have been possible without the implementation of ML.

In physical sciences, ML-based techniques are becoming more and more widespread (see [6] for an extended review). Besides unlocking completely new innovations, ML techniques have been utilized in situations where finding solutions to physical problems requires a lot of computing effort using conventional methods. For example, in fluid simulations where normally one is required to solve the Navier–Stokes equations, ML can be used to accurately predict the evolution of a fluid simulation [7], while reducing the amount of computation time of these complex simulations significantly. Furthermore, neural networks have been used to simulate light scattering by multi-layer nanoparticles and the subsequent design of these nanoparticles using backpropagation [8]. In spectroscopy, machine learning is used to accurately classify physical objects based on noisy/complex spectroscopic data [9–11]. Another field where ML techniques find their way is in experiments with extremely high inference rates and low latency constraints, such as in high-performance detector triggers [12] (e.g. the ATLAS experiment at the LHC at CERN). In these

experiments, field-programmable gate array (FPGA) implementations of ML techniques are used to create complex hardware triggers for events with sub-microsecond lifetimes.

However, ML techniques are notoriously opaque, meaning that, although they deliver desired results, our understanding of how the results are obtained and how patterns in data are detected by the underlying neural network is very limited [13]. Often, little or no information on the physical system that is solved/simulated can be gained by studying the neural network itself, which is a significant drawback of these techniques. Furthermore, basic questions regarding the minimal amount of training data, or the optimal size of the underlying neural network required to obtain good precision for a pre-defined problem remain unanswered [6].

Signal-parameter estimation is an integral part of both fundamental and applied research, with precision, accuracy and speed being crucial for the real-time observation and control of physical and chemical dynamic processes. A multitude of research fields rely on determination of time constants and frequencies of decaying signals, examples of which include: nuclear magnetic resonance (NMR) [14] where molecular structures are analysed from precise determination of the frequency and decay time of the Larmor precession of nuclear spins in a magnetic field; free-induction-decay (FID) optical magnetometry [15–19] where measurement sensitivities depend on the precision of the measurement of the precession of magnetized spins about a magnetic field; and cavity ring-down spectroscopy (CRDS) [20–24] that relies on the detection of variations of the photon lifetime in optical cavities to identify trace gasses, measure absorption cross-sections or observe chemical reactions in real time. Other cavity-enhanced methods, such as cavity ring-down polarimetry (CRDP) [25–31] and ellipsometry (CRDE) [32–34], measure birefringence and/or dichorism of an optical medium through the precise estimation of the signal-decay time and the polarization beat oscillations superimposed on such decaying signals.

All the aforementioned methods require some form of data processing to determine their relevant signal parameters. Optimizing sensitivity and minimizing computational costs is usually done by averaging multiple, consecutive measurements, sacrificing temporal resolution. Therefore, it is imperative that one considers the sampling and acquisition rates of an instrument and balances that against the computational costs (calculation time) to analyse the acquired data, the time scales of relevant dynamics to be observed, and the requirement to observe those dynamics in real time. In cavity-enhanced spectroscopy, CRDP/CRDE techniques in particular, the decay times of interest are typically in the 10^{-7}–10^{-5} s range, a few orders of magnitude smaller than the typical decay times of NMR experiments (10^{-2}–10^{1} s) or FID optical magnetometry techniques (10^{-3}–10^{1} s). As such, analysis methods that are sufficiently fast for the real-time analysis of single measurements of NMR or FID magnetometry experiments do not have the capability to offer real-time analysis in situations where the relevant time scales are much smaller than 1 ms. Different time- and frequency-based computational methods for rapid parameter estimation have been demonstrated, and evaluated, for cavity ring-down (CRD) spectroscopy methods [35–37]. Notably, Fourier-transform implementations on FPGAs have demonstrated data analysis rates as high as 4.4 kHz [38]. Several works discuss time- and frequency-domain analysis algorithms for damped sinusoidal signals [39, 40] and a recently published comparative study includes three analysis methods of discretely sampled damped sinusoidal signals in terms of their speed, and attainable accuracy and precision [41].

In this work we demonstrate a ML based approach to extract the relevant signal parameters from experimentally relevant signals with well-defined functional form. More specifically, we use dense autoencoder networks to encode, extract parameters from, and subsequently reconstruct two types of discretely sampled, decaying signals: (a) exponentially decaying signals and (b) decaying oscillations. We evaluate the autoencoder network on its precision and accuracy in parameter extraction and compare its performance to the fundamental estimation limits of such signals given by the Cramér–Rao lower bound (CRLB). We show that the dense autoencoder network is able to reach analysis rates of 75 kHz of 1000 sample signals with cost-effective computational facilities. This makes the ML method ideal for implementations where computational capabilities come at a premium, such as fast, portable cavity-enhanced sensing instruments.

2. Theory

2.1. Dense autoencoder neural networks

A neural network is an ordered group of neurons that, much like the neurons iDn brains, communicate by signaling to each other. Each neuron has a number of incoming and outgoing connections from and to other neurons. Each connection has a strength (weight) and can be stimulating or inhibiting the response of the receiving neuron. To calculate the activity (value) of a neuron, one compares the sum of its weighted inputs to a neuron specific reference (bias). The difference is passed through an activation function which produces the output of the neuron [1, 2]. Initially, the weights and biases within a neural network start off randomly.

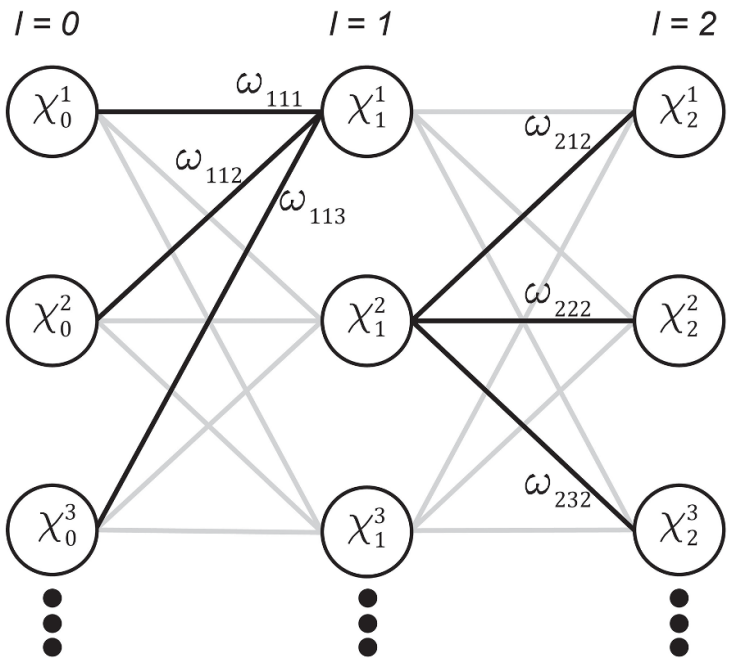

Figure 1. Example of a dense neural network with $L = 2$ layers. The first layer, $l = 0$ is the input layer, $l = 1$ a hidden layer and $l = 2$ is the output layer. χ_y^x specifies neuron x in layer y. Each neuron contains a bias and layer specific activation function (not illustrated). $\omega_{y,x,z}$ denotes the weight (strength) between neuron x in layer y and neuron z in layer $x-1$. For illustrative purposes, some connections have been highlighted and are shown with their corresponding weight. The output of the neuron depends on its inputs, weights and bias put through the activation function (equation (1)).

As the neural network is trained, these weights and biases are altered to optimize the performance of the network with respect to the task it is asked to perform. The manner in which neurons are ordered and connected influence the capabilities of the overall network and a variety of different network configurations with different purposes have been demonstrated [42]. A dense neural network is a network where the neurons are arranged in sequential layers and all the layers are fully connected. This means that every single neuron in one layer is connected to all neurons in the next layer. There are no neural connections within a layer or connections spanning multiple layers. An example of a dense neural network is shown in figure 1. For such a neural network, one can calculate its output as:

$$\begin{aligned} &\text{for } l : 0 \longrightarrow L \\ &\vec{\chi}_{l+1} = \mathcal{F}_l\left(\boldsymbol{\omega}_{l+1} \cdot \vec{\chi}_l - \vec{b}_l\right), \end{aligned} \tag{1}$$

where L is the number of layers in the neural network (excluding the input layer), $\vec{\chi}_l$ is the output of the network at layer l, $\boldsymbol{\omega}_l$ is the weights matrix, and $\vec{b}_l$ is the bias vector for the corresponding neurons of layer l. Finally, $\mathcal{F}_l$ is the activation function of the network layer that acts piece-wise on each neuron in layer l. The input layer ($l = 0$) has no bias or activation function.

Autoencoders are symmetric, hourglass-shaped neural networks [43–47] (figure 2). This means that the number of neurons in the middle layer of the network is small compared to the number of neurons in the input and output layer of the neural network. The goal of the autoencoder network is to recreate its input signal. In the first half, the autoencoder network learns to encode the input data into a lower dimensional representation. In the second half, the autoencoder network learns to subsequently decode that lower dimensional representation and reconstruct the original signal with minimal losses. The middle layer of the network, in which the maximal compression of the signal takes place, is called the latent space. The latent representation of the original signal can be extracted from the latent space. Autoencoder networks are typically used in data compression [46] and de-noising [47].

The choice of dimensionality of the latent space is '*problematic*' [44]. In order to achieve maximum data compression, the number of neurons in the latent space of the autoencoder network should be as small as possible. However one would also like to be able to reconstruct the data with minimal error. If the inherent dimensionality of the data is not known *a priori* (as is typical) [44, 46, 47] but important, sparse, or variational autoencoder networks with Bayesian regularization methods should be used [48]. Furthermore, inference from the latent space representation is nontrivial because the way a network 'learns' to encode a

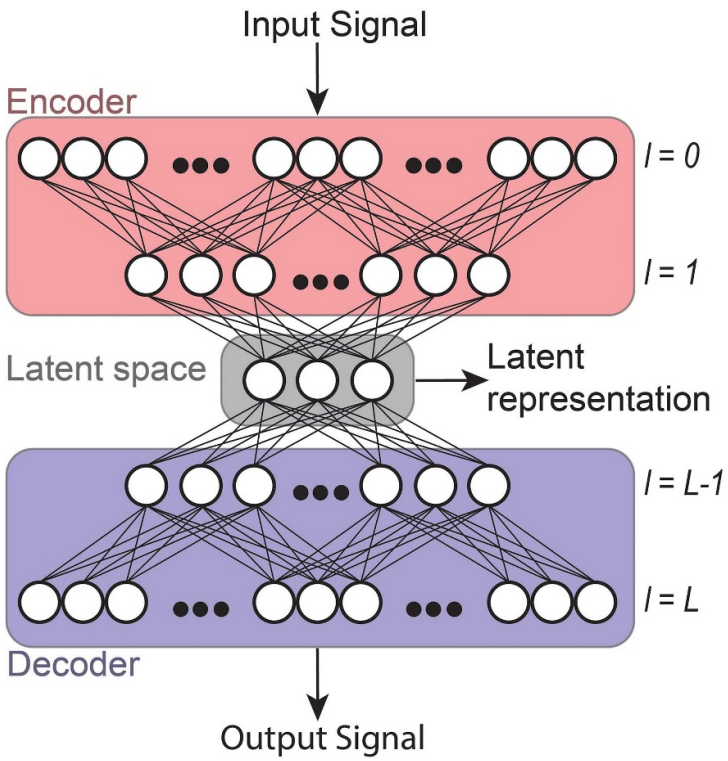

Figure 2. Example of an autoencoder neural network with L layers. Signals are fed to the encoder, which compresses the data into a limited number of parameters after which the decoder reconstructs the original data from the encoded parameters whilst minimizing the losses. The layer with the smallest number of neurons is called the latent space. From the latent space the encoded, or latent, representation of the input signal can be extracted.

signal in training is not necessarily unique. Another way of structuring the latent space of an autoencoder network is by including the Kullback–Leibler (KL) divergence during training. However, KL divergence inevitably introduces a trade-off between the ability of the autoencoder network to reconstruct the original signal and a structured latent space [49].

In this work, we examine how an autoencoder network consisting solely of dense layers can be employed for direct extraction of the parameters of a signal with well-defined functional form from its latent signal-representation, as this becomes relevant in real-time signal analysis. To achieve such a task, we must make sure that (a) the dimensionality of the latent space matches the dimensionality of the input signal, and (b) that the autoencoder network encodes the input signal in a specific way where the latent representation of the signal matches the independent parameters of the signal.

We are able to satisfy the first requirement because the number of parameters in our signal's functional form are known, which allows us to match the number of neurons in the latent space of the autoencoder network to the number of parameters of the function (sections 2.2 and 3.2). Thereby we achieve efficient encoding without redundant neurons in the latent space, or redundant values in the latent representation of the signal. We are able to satisfy the second requirement by training the autoencoder network in a specific, three stage training method (section 3.4).

2.2. Model signals

We use autoencoder networks to estimate the parameters of two types of experimentally relevant model signals: (a) exponentially decaying signals and (b) decaying oscillations.

2.2.1. Exponentially decaying signals

We start by investigating purely exponentially decaying signals. Such signals are encountered in a wide range of spectroscopic techniques, such as cavity ring down spectroscopy and cavity-enhanced sensing methods [22, 23, 50] and time-resolved fluorescence spectroscopy [51, 52]. An exponentially decaying signal can be characterized in terms of a model function as:

$$y(t) = A_0 \cdot e^{-t/\tau} + y_0(t), \tag{2}$$

where A_0 is the initial amplitude and τ is the decay constant of the signal, $y_0(t)$ is the signal offset, and t is the independent (time) variable that is discreetly sampled. Here, for simplicity and without loss of generality, we assume that $A_0 = 1$ and $\langle y_0(t)\rangle = 0$. Furthermore we restrict the investigation of noise contributions to the global offset parameter, i.e. $y_0(t)$. We model the noise to be normally distributed, i.e. $\langle y_0(t)\rangle = 0$, $\langle y_0^2(t)\rangle = \sigma_{y_0}^2$ and define the signal-to-noise ratio as: SNR $= A_0/\sigma_{y_0}^2 = \sigma_{y_0}^{-2}$. Under realistic experimental conditions, signal-amplitude fluctuations can be incorporated into the SNR through A_0. This way, the dimensionality of the exponentially decaying signal is reduced to a single parameter that in most cases of interest carries the valuable information: the decay constant τ.

2.2.2. Decaying oscillations

The second type of model signals we investigate are decaying oscillations, as these become relevant in a wide range of experimental techniques, particularly within NMR [14], optical magnetometry [15–19] and CRDP [25–31, 53] and CRDE [32–34, 54]. An exponentially decaying oscillation can be characterized in terms of a model function as:

$$y(t) = A_0 \cdot e^{-t/\tau} \cdot \cos(2\pi \cdot f \cdot t + \phi) + y_0(t), \tag{3}$$

were t is the discretely sampled independent variable of the signal (time), A_0 is the oscillation's initial amplitude, τ is the decay constant of the signal's envelope, f is the frequency of the oscillation, and ϕ its phase. Finally $y_0(t)$ is the global signal offset. Again, we assume the signals amplitude to be normalized ($A_0 = 1$) and no global signal offset ($\langle y_0(t)\rangle = 0$), and SNR is defined as SNR $= \sigma_{y_0}^{-2}$ identical to the exponentially decaying signals. The experimentally relevant parameters τ, f and ϕ, are the free parameters we extract from the latent representation. We recognize that, similar to the case of pure exponential decays, the amplitude parameter A_0 could potentially be an experimentally relevant parameter for inspection purposes, and could be considered as a free parameter of the model signal.

2.2.3. Cramér–Rao lower bound

The fundamental limits for the statistical uncertainties of determining the decaying time-constant and oscillation frequency of pure and oscillating decaying signals are described by the CRLB. For the case of a decaying (oscillating) signal (equations (2) and (3)) the CRLB [16, 55] sets the lower limit on the variance of both the decay constant estimator σ_τ^2 and the frequency estimator σ_f^2. In general, the CRLB limit can be defined for any parameter extracted by an unbiased estimator. Here however, we focus on the two most important parameters for the experimental techniques of interest, e.g. [14–23, 25–34, 50–54].

The relation between the variance limit of the decay-time estimator and the lower limit of the frequency estimator is given by: $\sigma_\tau^2 = 2\pi\sigma_f^2$. The CRLB for the frequency estimator σ_f^2 is given by [16, 55]:

$$\sigma_f^2 = \frac{6}{(2\pi)^2\,\mathrm{SNR}^2\,\mathrm{f_{BW}}\,T_\mathrm{m}^3}\,\xi(\tau/T_\mathrm{m}), \tag{4}$$

where SNR is the signal-to-noise ratio of the decaying oscillating signal; f_{BW} is the sampling-rate-limited bandwidth of the measurement; T_m is the measurement time window and, $\xi(\tau/T_\mathrm{m})$ is a correction factor that takes into account the signal decay, which is given by [16, 55]:

$$\xi(r) = \frac{\exp(2/r) - 1}{3r^3\cosh(2/r) - 3r(r^2+2)}. \tag{5}$$

The factor $\xi(\tau/T_\mathrm{m})$ serves as a compensation factor in equation (4) that penalizes measurement of the tails of the exponential decay when the signal has effectively died out. The validity of equation (4) hinges on the assumption that the period of the oscillation is shorter than the typical decay time (τ) of the signal envelope, and that a sufficient number of oscillations occur in the measurement time window. Moreover, equation (4) dictates that any noise sources affecting the signal detection are contributing to the fundamental CRLB limit through their effect on the SNR of signal.

In [30], it was demonstrated that the CRLB limit is the appropriate estimator of the fundamental sensitivity of frequency-based measurements within the context of cavity-enhanced spectro-polarimetric techniques [25–34, 53, 54].

3. Methods

3.1. Simulated signals and data-sets

We use simulated signals generated using the model functions presented in equations (2) and (3) to train and demonstrate the parameter-extraction capabilities of dense autoencoder networks. For our simulated signals—both pure exponentially decaying and decaying oscillating ones—we assume a total signal duration of 5 μs and an effective bandwidth-limited sampling rate of 200 MHz, resulting in discretely sampled signals with a length of 1000 samples. Such signals are frequently encountered in CRDP experiments [30]. We generate training data consisting out of multiple signals with varying experimentally relevant parameters. From the training data the autoencoder network learns the relationship between the shape of the signal and the values of the relevant parameters. For exponentially decaying signals (equation (2)), the only parameter of interest is the decay constant τ, which we vary between simulated signals as follows:

$$\tau = \left|\mathrm{norm}\left(\mu_\tau = 1\ \mu\mathrm{s}, \quad \zeta_\tau = 0.5\ \mu\mathrm{s}\right)\right|, \tag{6}$$

where μ_τ and ζ_τ are the average and standard deviation of a normal distribution, from which we take the absolute value as a negative decay constant would lead to exponentially increasing signal. For the decaying oscillations (equation (3)) we vary the decay constant using the same process, but also vary the other two experimentally relevant parameters, the frequency and phase of the oscillation, f and ϕ respectively, using:

$$f = \mathrm{norm}\left(\mu_\mathrm{f} = 3\ \mathrm{MHz}, \quad \zeta_f = 0.1\ \mathrm{MHz}\right) \tag{7}$$

$$\phi = \mathrm{norm}\left(\mu_\phi = 0, \quad \zeta_\phi = 0.1\right). \tag{8}$$

For the exponentially decaying signals a training data set consists of 200 signals, while for the decaying oscillations, a training data set consists of 1000 signals. These signals have a SNR $= 2^{20}$, to allow the autoencoder network to learn the features of each model signal and the signals variance under changing underlying parameters of interest without being hindered by noise artefacts.

We would like to mention here that, in real-world applications of this analysis method, the parameter range on which the autoencoder network is trained is of crucial importance to the functionality of the method. The training parameter range should correspond to the expected experimental parameter range. Therefore, prior knowledge about the parameters and the expected range of these parameters is required for the autoencoder to analyse experimental signals. However, in a real-world experiment, or commercially available instrument, baseline signal parameters will be well defined and within well specified ranges making the analysis method viable.

3.2. Dense autoencoder networks

The autoencoder networks that we use to reconstruct and extract the parameters from decaying signals, have an input layer and output layer of 1000 neurons, equal to the number of samples in a signal. However the number of neurons in the latent space of the autoencoder networks are different between autoencoder networks that analyse exponentially decaying signals and decaying oscillations. In the case of the exponentially decaying signals, we use a network with only a single neuron in the latent space, corresponding with the single parameter of interest that we wish to extract from the signal, i.e. τ. In the case of the decaying oscillations, we need three neurons in the latent space of the network in order to be able to extract the three interesting parameters of that signal, i.e. τ, f, ϕ. Furthermore, the signal complexity of a decaying oscillation with three parameters is higher than a exponentially decaying signal with only one parameter of interest. For this reason, to analyse decaying oscillations we increase the number of neurons in the layers of the encoder and decoder part of the autoencoder network (figure 2). The network we use to analyse the exponentially decaying signal has five total layers with 1000–50–1–50–1000 neurons per layer respectively. For the decaying oscillations, we use a network with seven layers and 1000–50–10–3–10–50–1000 neurons per layer respectively. We choose to use a hyperbolic tangent as activation function for all relevant layers. This choice allows the neural network to express values between -1 and 1, which matches the maximal amplitude of the input and output signals. We create the aforementioned neural networks using the Tensorflow and Keras [56] libraries in a homemade Python script.

3.3. Signal reconstruction and parameter extraction

Autoencoder networks are used to reconstruct the original input data. However, our focus here is to use autoencoder networks for the rapid extraction of the signal parameters from exponentially decaying signals and decaying oscillations from the latent space of the autoencoder network. To do this we map the signal

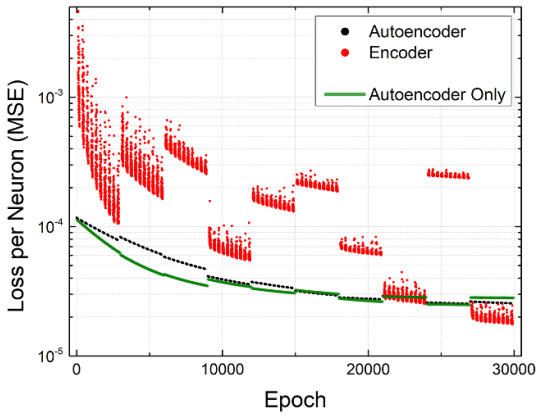

Figure 3. Average MSE loss per neuron as a function of training epochs for two networks. One network is trained using the three-stage training method outlined in section 3.4 (dots) while the second network is trained in a typical fashion for an autoencoder network consisting of the first step only (dark-green line). The two networks have the same shape and size, are trained for an equal number of epochs using an equal amount of training data. The black and red dots correspond to the losses of the first network at training steps 1 and 2 respectively (see text). The losses at training step 3 are omitted from the figure because they overlap with the losses of the first training step. The breaks in the losses of the encoder network training (red dots) when a new data set is generated are attributed to the small number of neurons in the latent space of the autoencoder network and the specifics of the small validation data set.

parameters of interest to a number between −1 and 1 that the autoencoder network can express in the latent space:

$$x_{\text{lat}} = \frac{x - \mu_x}{3 \times \zeta_x}, \tag{9}$$

where x_{lat} is the latent representation of the value of parameter x (τ, f, and ϕ). μ_x and ζ_x are the mean and standard deviation of the parameter variation defined in equations (6)–(8). We use equation (9) to create a desired latent representation of signal parameters during training, and to convert the latent representation back into the signal parameter when using the autoencoder network to analyse signals. For a trained neural network, a signal only needs to be passed through the encoding part of the network, up to the latent space, in order to find its latent representation. To investigate the analysis speed of the neural networks, we only take into account the time it takes to encode the original signal into the latent representation.

3.4. Training protocol

The autoencoder networks are trained following a three-stage training scheme, using a stochastic gradient descent (SGD) algorithm [56] and use a mean squared error loss function given by

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{10}$$

where n are the total number of samples in a signal, y is the desired signal or representation and $\hat{y}$ is the predicted outcome by the autoencoder network. Even though the sizes of the autoencoder networks are different given the two different model signals, the training method for the autoencoder networks is the same:

(a) The complete autoencoder network is trained to recreate simulated input signals for 100 epochs.
(b) The encoder part of the autoencoder network, up to the latent space (figure 2), is trained to generate the desired latent-signal representation from simulated input signals.
(c) Finally, the decoder part of the autoencoder network is trained with the desired latent-signal representation as input and the desired signals as output for 100 epochs.

We repeat these steps ten times after which a new training data set is generated to avoid over-fitting a single data set. In total, each network is trained on ten different data sets. By implementing three distinct training steps we are able to train the neural network to not only recreate the original signal, but also to encode the signal in a specific way that allows extraction of the signal parameters from the latent space.

Figure 3 compares the prediction error (loss) of two autoencoder networks during training. We measure the loss of each neuron in the output layer and the latent space using the mean squared error (MSE) [56]. The smaller the loss of the autoencoder network, the better the network is in recreating the desired output.

The first network (red and black dots) is trained using our three-stage training method. The second network (dark-green line) is only trained on replicating the input data (first step). Both networks encode and reconstruct exponentially decaying signals described by equation (2). We observe that both training methods converge to similar losses per neuron at the end of training for the complete autoencoder networks (black dots versus dark-green line), meaning that both networks, at the end of training, are able to reproduce the input data with a similar level of precision and accuracy. However, in the autoencoder network that is trained using our three-stage training method, we see a decrease in losses from the encoder part of the autoencoder network (red dots), indicating that by using the three-stage training method we can achieve a desired latent signal representation without sacrifice of the capability of the autoencoder network to reconstruct the original signal.

4. Results

4.1. Exponentially decaying signals

Figure 4 shows the results for a trained autoencoder network reconstructing and extracting parameters from exponentially decaying signals. The ability of the autoencoder network to extract the signal parameters from the latent space is compared to the performance of an ordinary least squares algorithm [30]. In figure 4(a), we show an example of a generated exponential decaying signal with SNR $= 2^5$ and a randomly selected time constant ($\tau = 1.81$ μs). We use a trained autoencoder network (section 3.4) to reconstruct this generated signal. In addition, we use a least-squares algorithm to analyse the generated signal, from which we extract the expected decay constant $\tau = 1.81(2)$ μs, with a precision in accord to the CRLB limit (see related discussion in [41]). In figure 4(b), we show a histogram of 500 estimates of the decay constant extracted from the latent space of the autoencoder network. By subtracting the least squares decay constant estimate (τ_{est}) and fitting the histogram to a Gaussian distribution we show that there is no difference in accuracy between the two analysis methods as the center of the distribution is not significant from zero. The width of the distribution of the decay constant estimates coincides with the uncertainty of the least squares estimation method indicating that also the precision of the two methods is equal. In figure 4(c), we demonstrate that the trained autoencoder network is able to, accurately and without loss in precision, follow signal changes. In particular, we simulate a spectral absorption feature having a characteristic dispersive (Lorentzian) profile. In this case, the decay constant changes by 30% and both the autoencoder network and the least squares algorithm are able to accurately detect such a change. This situation is representative of a spectroscopy experiment investigating, for instance, absorption from gaseous species using CRDS [24, 57]. In figure 4(d), we show that the precision of the neural network is able to match the precision of the least squares algorithm and that both are limited by the CRLB over several orders of magnitude in SNR. It is crucial to emphasize that the network is able to accurately and precisely extract the correct signal parameters over a wide range of SNR values, despite the fact that we do not vary the SNR of the input training data during the training of the network.

4.2. Decaying oscillations

In figure 5, we present the results of a trained autoencoder network reconstructing and extracting parameters from decaying oscillations. In figure 5(a), we show an example of a decaying oscillating signal with SNR $= 2^5$, $\tau = 1.28$ μs, $\phi = -0.243$, and $f = 2.972$ MHz. We fit the signal to equation (3) using a least squares algorithm and use a trained autoencoder (section 3.4) to reconstruct the signal. Figure 5(b) shows the histograms of 500 parameter estimates of τ , ϕ and f, extracted from the latent space of the autoencoder network. We extract the latent parameters $\tau_{\text{lat}}, \phi_{\text{lat}}, f_{\text{lat}}$ from the latent space of the autoencoder network and use equation (9) to reconstruct the estimate of the signal parameters found by the autoencoder $\tau_{\text{enc}}, \phi_{\text{enc}}, f_{\text{enc}}$. Following the same procedure described for the case of pure exponentially decaying signals, for each parameter we subtract the parameter's estimated value obtained through the least squares method and fit the histograms to a Gaussian distribution. Each distribution has a center that is not significant from zero and a width equal to the uncertainty of the fit parameters found by the least squares fit indicating that there is no difference in accuracy or precision between the two analysis methods. In figure 5(c), we demonstrate that the trained autoencoder network is able to, accurately and without loss in precision, follow signal changes. In particular, in figure 5(c), we simulate a spectral feature having a characteristic change in both its absorption and dispersion (e.g. Cotton effect). This results in signals whose frequency and decay constant change in a correlated way. Finally, in figure 5(d), we show that the precision of the parameter estimates from the latent space of the trained autoencoder network follows the precision of the least squares method over five orders of magnitude in SNR. Moreover, the precision of the parameters extracted from the latent space of the autoencoder network approaches the fundamental estimation limit as given by the CRLB.

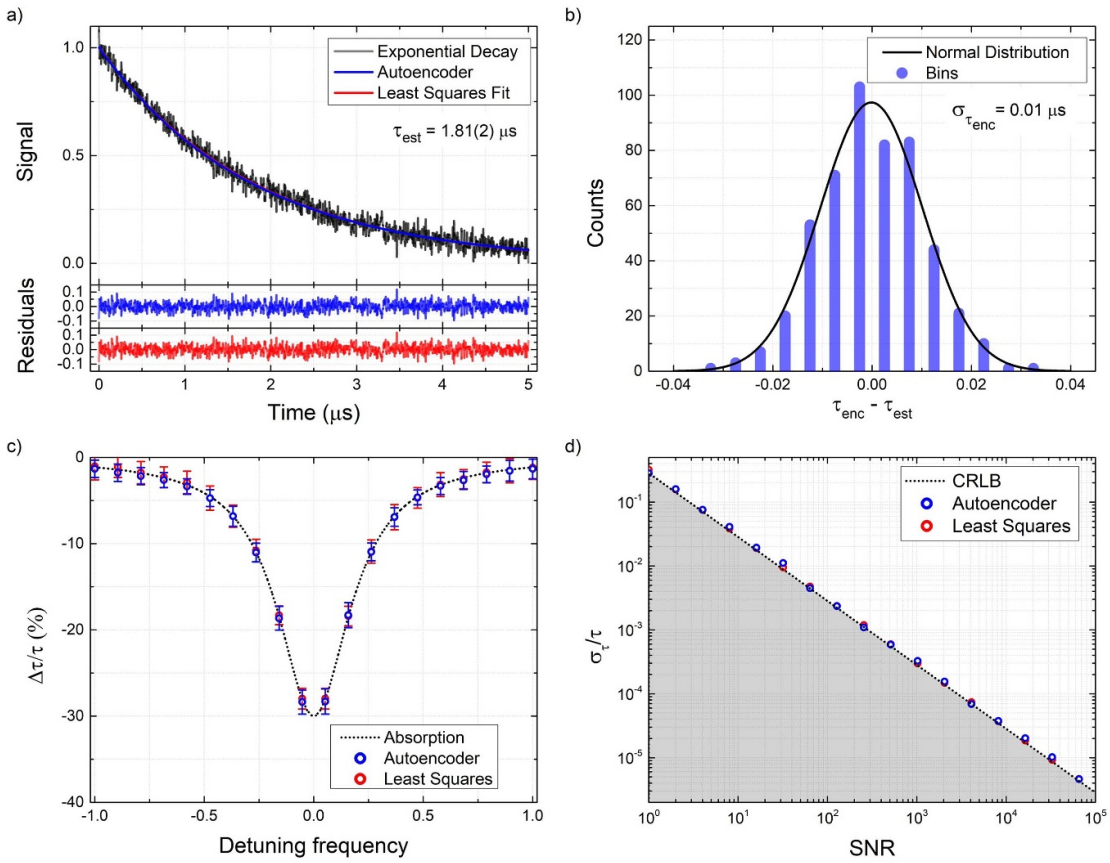

Figure 4. Parameter extraction from an exponentially decaying signal using an autoencoder network. (a) Example of an exponentially decaying signal with 1000 samples and SNR= 2^5 (black line) reconstructed by a trained autoencoder network (blue line) and analysed using a least squares fitting method (red line). The residuals for each method are shown in the bottom insets of the figure with their corresponding colour. The decay constant estimated by the least squares method is $\tau_{est} = 1.81(2)\,\mu s$. (b) Histogram of 500 decay-constant estimates extracted from the latent space of the trained autoencoder network. Analysed signals have decay constants and SNR equal to the signal shown in (a). The value of the decay constant estimated by the least squares method, τ_{est}, is subtracted from the decay constant extracted from the latent space of the autoencoder network, τ_{enc}. The full width at half maximum (FWHM) of the fitted Gaussian distribution coincides with the uncertainty of the least squares method while the center of the distribution is not significantly different from 0. (c) Accurate and precise signal parameter extraction using an autoencoder network is possible in the case of (relative) changes in the decay constant through a simulated absorption feature (dotted line) as one would expect in, for instance, cavity ring-down absorption spectroscopy when a laser is detuned from the resonance frequency of the absorption feature. The feature is followed by both the trained autoencoder network (blue dots) and a least squares algorithm (red dots). The error bars represent the standard deviation of 100 parameter estimations from the latent space of the trained autoencoder network and the least squares algorithm respectively. (d) Precision of a least squares algorithm and a trained autoencoder network on estimating the decay constant from the latent space over multiple orders of magnitude in SNR. The CRLB giving the fundamental estimation limit on the decay constant is also shown (dotted line) and the area below the fundamental estimation limit is greyed out.

4.3. Complexity vs calculation time

Analysis speed is crucial for the real-time investigation and control of fast processes. Faster methods of analysis allow for large data stream to be analysed quicker, driving down computational costs.

For our particular cases of interest, we recently showed [41] that non-iterative FFT methods require ∼300 μs of calculation time whereas a least-squares fitting algorithms requires >1 ms of calculation time under identical signal conditions as we present in this work. Similarly, other works have shown that FPGA-based systems running FFT algorithms are able to determine the decay constant of exponentially decaying signals with analysis rates of 4.4 kHz [38]. Here, we show that trained autoencoder networks are able to analyse both exponentially decaying signals and decaying oscillations significantly faster than any previously reported method. We compare the calculation speed of the encoder part of trained autoencoder networks of different sizes on different systems: a central processing unit (CPU) and a graphics-processing unit (GPU) based system. The CPU-based system is based around an Intel(R) Xeon(R) W-2123, 3.6 GHz processing unit with access to 16 GB of random-access memory (RAM) with a frequency of 1330 MHz, while the GPU system is based on a NVIDIA Quadro p5000 graphics board.

In figure 6, we present the calculation time required by the encoder part of an autoencoder network to analyse a single decaying oscillation signal for varying sizes of networks. For our estimation of the overall

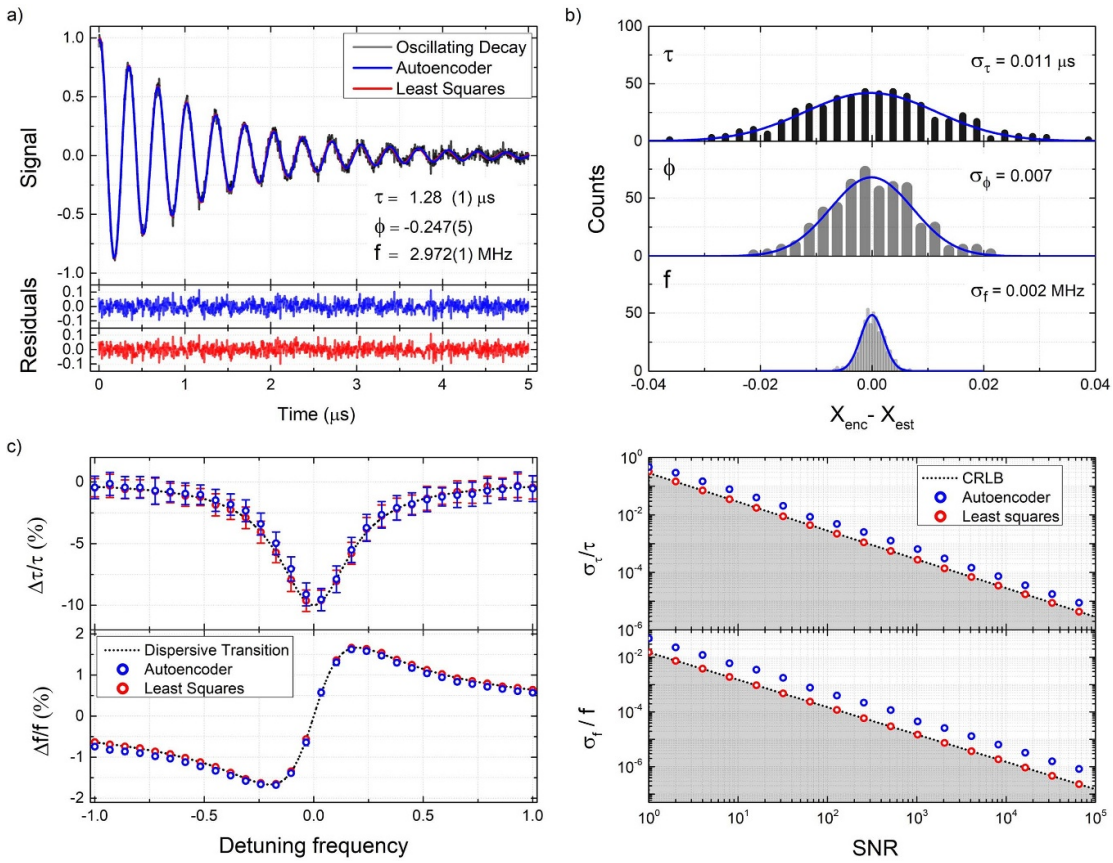

Figure 5. Parameter extraction from decaying oscillating signals using an autoencoder network. (a) Example of a decaying oscillation (black line) with SNR $= 2^5$ reconstructed by a trained autoencoder network (blue line) and analysed using a least squares fit (red line). The residuals for each method are shown in the bottom insets with their corresponding colour. The signal parameters found by the least squares fit are shown in the figure with their corresponding uncertainties. (b) Histograms of 500 signal parameter sets (top: τ, middle: ϕ and bottom: (f) extracted from the latent space of the autoencoder network. The analysed signals have the same parameters and SNR as the example signal shown in (a). For each of the signals parameters, we subtract the parameter estimate of the least squares fit X_{est} from the parameter estimate extracted latent space of the autoencoder network X_{enc}. The width of the fitted Gaussian distributions coincide with the uncertainty of the least squares method for each individual parameter of the signal and the centers of the distributions are not significantly different form 0. (c) Accurate and precise signal parameter extraction using an autoencoder network is possible in the case of (relative) changes in the decay constant through a simulated disperive absorption feature, or cotton effect, (dotted line) as one would expect in a cavity enhanced polarimetry experiment when a laser is detuned from the resonance frequency of the absorption feature. The feature is followed by both the trained autoencoder network (blue dots) and the least squares algorithm (red dots).The error bars represent the standard deviation of 100 parameter estimations from the latent space of the trained autoencoder network and the least squares algorithm respectively. (d) Precision of the trained autoencoder network (blue dots) and the least squares algorithm (red dots) in estimating both the frequency (f) and decay constant (τ) over several orders of magnitude in SNR. The fundamental estimation limits of these parameters, given by the CRLB is also shown (dotted line) and the area below the fundamental estimation limit is greyed out.

CPU calculation time (figure 6) we count the matrix multiplications, the adding/subtracting of the network biases and the application of the activation function. Other prepossessing operations for the CPU-based system, such as data collection, take between 3.6 and 4.9 μs per signal and are not included. For our estimation of the overall GPU calculation time we include the aforementioned operations (such as matrix multiplications, the adding/subtracting of the network biases, and the application of the activation function) and the transfer time of the results of the encoder onto the RAM of the computer. We do not include, however, the time it takes for the original signal to be transferred from the RAM onto the GPU memory ($\sim$5 μs) or any other pre/post-processing operations conducted by the CPU of the system ($\sim$6 μs). For the GPU-based system there is a clear difference in calculation time between neural networks with different numbers of layers in the encoder network, something we do not observe for the same neural networks on the CPU-based system. The results we present in figures 4 and 5 are obtained with autoencoder networks that require the calculation of $\sim$100 000 floating point operations (FLOPS) to reach the latent space representation of the signal. Using the trained autoencoder networks we are able to achieve analysis rates upwards of 75 kHz using the GPU-based system. This includes the data transfer time, the pre/post processing

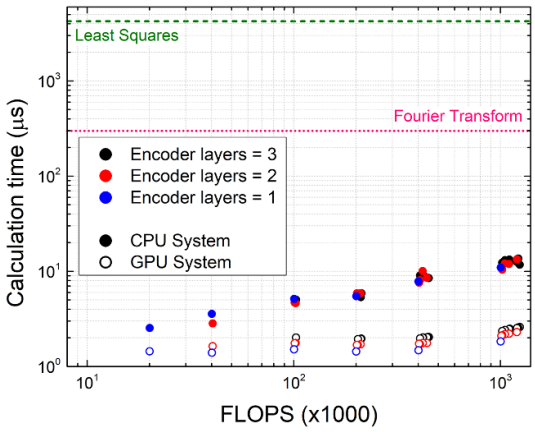

Figure 6. Calculation time required by the encoder part of the autoencoder network to extract the latent parameters of a single damped oscillating signal as a function of floating point operations (FLOPS) required to calculate the network using a CPU (solid dots) and a GPU (open dots) based system. The analysed signals consist of 1000 samples. The networks are varied in size, and thereby the required number of FLOPS needed to calculate latent signal representation. We vary the size of the network by varying the number of layers in the encoder network and the number of neurons within each layer. The lengths of these layers range from 10 to 500 neurons, where the smallest network investigated has a single layer with 10 neurons, and the largest network has three layers with 500, 200 and 100 neurons between the input layer and the latent space respectively. The dashed and dotted horizontal lines represent the calculation times required to analyse the same signal using a least squares algorithm and non-iterative Fast Fourier transform method, respectively [41].

operations conducted by the CPU, and the actual calculation time required of the autoencoder network itself. In future work, optimization of the data transfer rate between the data-acquisition system and data-analysis system and minimization of preprocessing operations, down to a combined $\sim 3\ \mu$s, the method of analysis we propose in this paper could be shown to be capable of reaching analysis rates of >200 kHz, in which we include the $<2\ \mu$s calculation time required by the GPU system.

5. Outlook and conclusion

In summary, we have shown that we can accurately and precisely extract the signal parameters of decaying signals using simple autoencoder networks. We demonstrate that our approach is orders of magnitude faster than conventional algorithmic methods (e.g. least-squares or FFT), regardless of CPU or GPU implementation of the neural network. We demonstrate analysis rates upwards of 75 kHz for signals with 1000 samples, and illustrate that analysis rates of >200 kHz are feasible with further optimization of data transfer speed between a data-acquisition and data-analysis device, which would allow for real time signal analysis rates of >200 kHz using state-of-the-art GSa/s sampling rates. Such capabilities can enable the real-time signal analysis of, for instance, CRDP signals at >200 kHz rates using state-of-the-art acquisition modules that have GSa/s sampling rates.

Concluding, we wish to note that the methodology of signal parameter extraction directly from the latent space of dense autoencoder networks could be applicable to other signal types that currently use fitting models for parameter extraction. Presently, neural networks are used for the classification of spectroscopic data [9–11], however, our approach can be directly implemented to analyse spectroscopic data for quantitative rather than qualitative results. Signals of higher complexity, such as signals with additional decay constants or frequency components, will require larger networks and larger training data sets for the network. If the number of extra parameters is known, a similar technique to what we demonstrate can be employed. However, if the number of additional parameters is unknown, a regularization method should be used to adequately choose the number of latent space parameters of the network to analyse the signal.

Data availability statement

The data that support the findings of this study are available upon request from the authors.

Acknowledgments

J C V and L B are grateful to Christian Schmitt and Michael Everest for their help and support and specially thank Sharon van Etten for insightful discussions. This work was supported by the European Commission Horizon 2020, Project ULTRACHIRAL (Grant No. FETOPEN-737071).

Appendix. Influence of realistic noise sources on autoencoder reconstructions and parameter estimations

Studying additional noise sources and their influence on the autoencoder networks signal reconstructions and parameter estimations can be highly informative, considering that real-world experiments encounter real-world noise. We consider frequency and amplitude noise as two additional noise sources and, in this section, study how these noise sources affect the signal reconstruction and parameter estimation differently compared to the offset noise discussed in the main text of this work. For this investigation we limit ourselves to decaying oscillations only.

Decaying oscillations with frequency noise are given by:

$$y(t) = A_0 \cdot e^{-t/\tau} \cdot \cos\left(2\pi \cdot f(t) \cdot t + \phi\right) + y_0(t), \tag{11}$$

where the frequency is given by

$$f(t) = f_0 + \mathrm{norm}\left(\mu, \sigma\right) \tag{12}$$

and $\mathrm{norm}\left(\mu, \sigma\right)$ is a normally distributed random process with mean μ and standard deviation σ. We compare autoencoder reconstructions and latent space frequency estimates of decaying oscillations under three distinct cases of frequency noise where:

(a) $\mu = 0$ kHz; $\sigma = 0$ kHz,
(b) $\mu = 0$ kHz; $\sigma = 30$ kHz,
(c) $\mu = 30$ kHz; $\sigma = 30$ kHz.

The first case gives a baseline estimate without frequency noise, the second case adds frequency noise to the signals, the third and final case we investigate has both frequency noise and a frequency offset of equal magnitude. In each case, we investigate the additional frequency noise (or lack thereof) on a baseline offset noise corresponding to a signal-to-noise ratio of 2^5, an equivalent noise level shown in figure 5.

Figures 7(a)–(c) show the autoencoder reconstructions and least-squares fits for the three cases of frequency noise. It can clearly be observed from the residuals of the autoencoder network reconstructions that, in the presence of frequency noise, the autoencoder is unable to fully reconstruct the signal (figures 7(b) and (c)). Figure 7(d) shows histograms of 500 frequency estimates of the cases of frequency noise and area-equalized Gaussian fits of said histograms. The estimates and uncertainties from the least-squares fits are also shown. The autoencoder network frequency estimate distributions in the cases with additional frequency noise are $\sim$20% wider compared to the case without frequency noise. Both cases where frequency noise is present pick up a bias compared to the real frequency of the decaying oscillating signal. Remarkably, the frequency estimates obtained form the least-squares fitting technique also pick up a slight, but significant bias in the presence of frequency noise. The width of the frequency estimates from the least-squares method also widens by $\sim$8%. The autoencoder network is able to differentiate between cases 2 and 3 where a frequency offset of the same magnitude as the frequency noise is applied.

Considering amplitude noise in decaying oscillating signals, we take the a similar and express signals containing amplitude noise as:

$$y(t) = A(t) \cdot e^{-t/\tau} \cdot \cos\left(2\pi \cdot f \cdot t + \phi\right) + y_0(t), \tag{13}$$

where the amplitude of the signal is given by

$$A(t) = A_0 + \mathrm{norm}\left(\mu, \sigma\right). \tag{14}$$

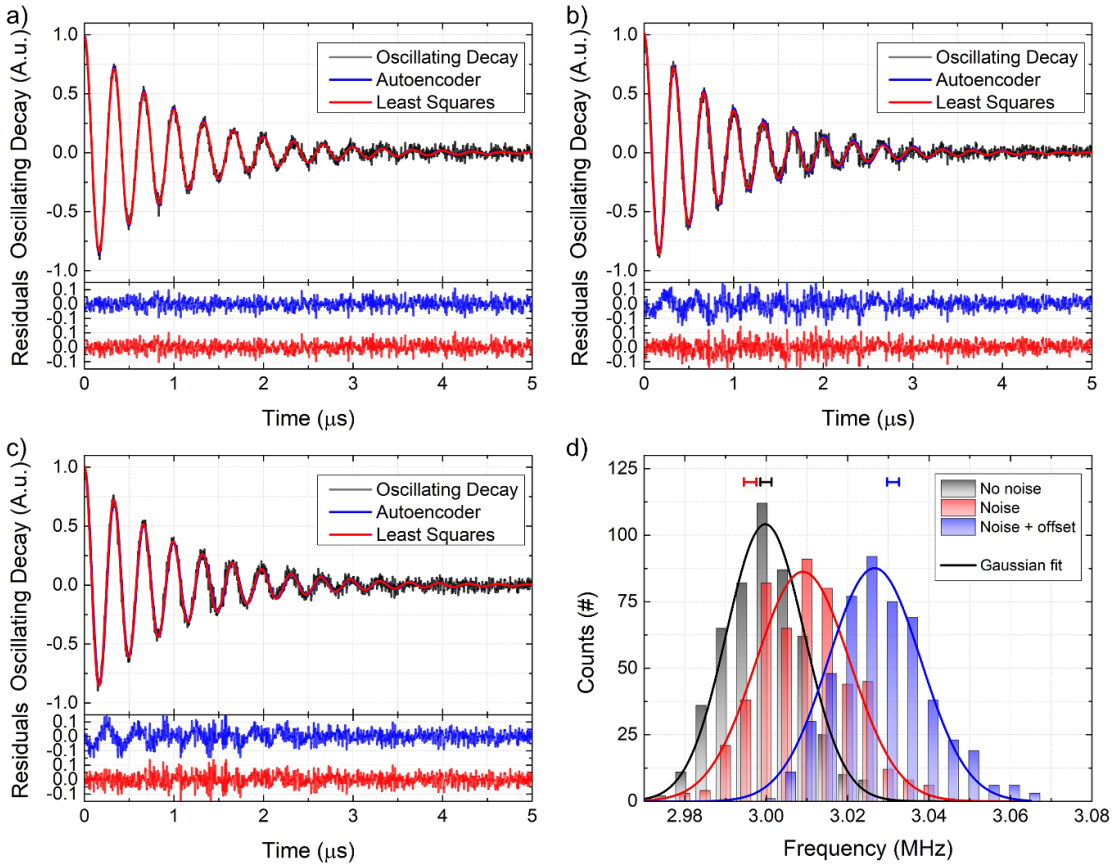

Figure 7. Autoencoder reconstructions and least squares fits of decaying oscillating signals with baseline SNR $= 2^5$ for the cases of (a) no additional frequency noise, (b) additional frequency noise with $\sigma = 30$ kHz, and (c) additional frequency noise and frequency offset of $\mu = 30$ kHz, $\sigma = 30$ kHz. The residuals for the autoencoder reconstruction and the least-square fit are shown in the bottom insets of each figure with their corresponding colour. (d) Histograms and area-equalized Gaussian fits of 500 frequency estimates of the three cases of frequency noise (black = no frequency noise, red = additional frequency noise, blue = additional frequency noise and frequency offset). The least-squares estimates are shown along with their respective uncertainty above the histograms.

Again we study three cases where:

(a) $\mu = 0$; $\sigma = 0$,
(b) $\mu = 0$; $\sigma = 0.03$,
(c) $\mu = 0.03$; $\sigma = 0.03$.

Figures 8(a)–(c) show the autoencoder reconstructions and least square fits for the three cases of amplitude noise. Interestingly, cases 2 and 3 show progressively more remaining signal in the residuals of the autoencoder reconstructions. In contrast, the least-squares method is able to reconstruct the signal effectively in all cases. Figure 8(d) shows area-equalized Gaussian fits from 500 parameter estimations of the respective amplitude noise cases and the estimates and uncertainties corresponding to the least-squares method. The least-squares method has no problem handling the amplitude noise cases and none of the parameter estimates differ significantly between the different cases of amplitude noise. It can be seen from figure 8(d) that the autoencoder network attempts to accommodate the amplitude noise with an increased decay time (top) and slightly retarded phase (middle). The frequency-estimate distributions for the cases with amplitude noise are slightly narrower ($\sim$8%) compared to the frequency-estimate distribution of the case without amplitude noise (bottom).

Overall, the addition of extra, realistic noise sources such as frequency and amplitude noise increases the estimation uncertainty of, and can introduce biases in, the parameter estimates of the autoencoder networks. However, the parameter estimates of the autoencoder networks do not completely deteriorate from the addition of these noise sources. The results shown in figures 4 and 5 of the main text were chosen as the most

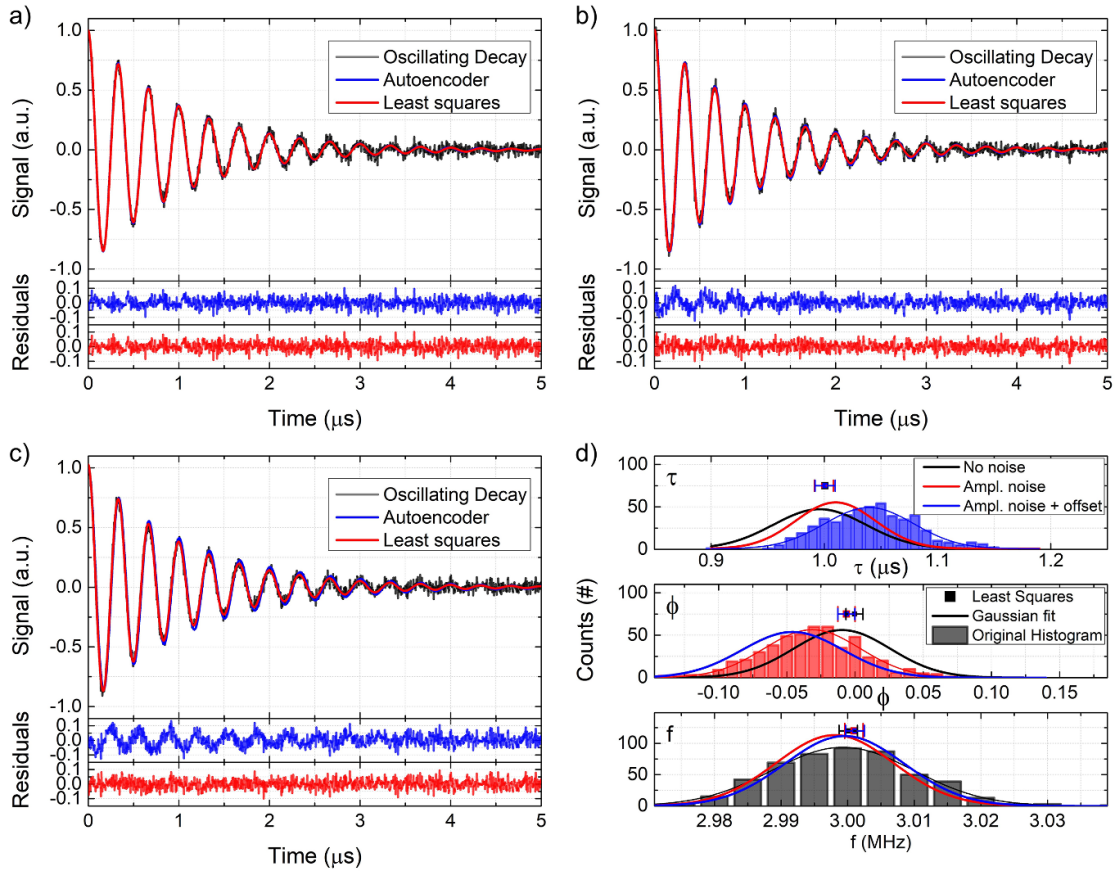

Figure 8. Autoencoder reconstructions and least squares fits of decaying oscillating signals with baseline SNR $= 2^5$ for the cases of (a) no additional amplitude noise, (b) additional amplitude noise with $\sigma = 0.03$, and (c) additional amplitude noise and amplitude offset where $\mu = 0.03$ and $\sigma = 0.03$. The residuals for the autoencoder reconstructions are given in the bottom insets of each figure with their corresponding colour. (d) Area-equalized Gaussian fits of histograms from 500 parameter estimates for the three noise cases shown in (a), (b), and (c). For illustrative purposes only one original histogram is shown in each subplot. The least-squares estimates and corresponding uncertainties are given in each subplot above the distributions.

general representation of a wide range of possible experiments ranging in SNR over five orders of magnitude (for offset noise). We would also like to note that the autoencoder network used to reconstruct and predict the parameters from signals affected by frequency- and amplitude noise are the same networks used to acquire the results presented in figure 5 in the main body of our work and have not been re-trained to accommodate these types of noise specifically.

ORCID iDs

Jim C Visschers https://orcid.org/0000-0001-5955-4858
Dmitry Budker https://orcid.org/0000-0002-7356-4814
Lykourgos Bougas https://orcid.org/0000-0002-8050-1141

References

[1] Bishop C M *et al* 1995 *Neural Networks for Pattern Recognition* (Oxford : Oxford University Press)
[2] Hertz J A 2018 *Introduction to the Theory of Neural Computation* (Boca Raton, FL: CRC Press)
[3] Gao C, Saraf A, Huang J-B and Kopf J 2020 Flow-edge guided video completion *European Conf. on Computer Vision* (Springer) pp 713–29
[4] Shamir A, Mitra N J, Umetani N and Koyama Y 2020 Intelligent tools for creative graphics *ACM SIGGRAPH 2020 Courses* pp 1–11
[5] Brown T B *et al* 2020 Language models are few-shot learners (arXiv:2005.14165)
[6] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
[7] Sanchez-Gonzalez A, Godwin J, Pfaff T, Ying R, Leskovec J, and Battaglia P 2020 Learning to simulate complex physics with graph networks *Int. Conf. on Machine Learning* (PMLR) pp 8459–68
[8] Peurifoy J, Shen Y, Jing Li, Yang Y, Cano-Renteria F, DeLacy B G, Joannopoulos J D, Tegmark M and Soljačić M 2018 Nanophotonic particle simulation and inverse design using artificial neural networks *Science Advances* **4** eaar4206

[9] Wang J, Liao X, Zheng P, Xue S and Peng R 2018 Classification of chinese herbal medicine by laser-induced breakdown spectroscopy with principal component analysis and artificial neural network *Anal. Lett.* **51** 575–86
[10] Del Moral F G, Guillén A, Del Moral L G, O'valle F, Martínez L and Del Moral R G 2009 Duroc and iberian pork neural network classification by visible and near infrared reflectance spectroscopy *J. Food Eng.* **90** 540–7
[11] Gniadecka M *et al* 2004 Melanoma diagnosis by raman spectroscopy and neural networks: structure alterations in proteins and lipids in intact cancer tissue *J. Investig. Dermatol.* **122** 443–9
[12] Nottbeck N, Schmitt C and Büscher V 2019 Implementation of high-performance, sub-microsecond deep neural networks on FPGAs for trigger applications *J. Instrum.* **14** 09014
[13] Burrell J 2016 How the machine 'thinks': understanding opacity in machine learning algorithms *Big Data Soc.* **3** 2053951715622512
[14] Günther H 2013 *NMR Spectroscopy: Basic Principles, Concepts and Applications in Chemistry* (New York: Wiley)
[15] Savukov I M and Romalis M V 2005 NMR detection with an atomic magnetometer *Phys. Rev. Lett.* **94** 123001
[16] Gemmel C *et al* 2010 Ultra-sensitive magnetometry based on free precession of nuclear spins *Eur. Phys. J.* D **57** 303–20
[17] Nikiel A, Blümler P, Heil W, Hehn M, Karpuk S, Maul A, Otten E, Schreiber L M and Terekhov M 2014 Ultrasensitive ^{3}He magnetometer for measurements of high magnetic fields *Eur. Phys. J.* D **68** 1–12
[18] Grujić Z D, Koss P A, Bison G and Weis A 2015 A sensitive and accurate atomic magnetometer based on free spin precession *Eur. Phys. J.* D **69** 135
[19] Hunter D, Piccolomo S, Pritchard J D, Brockie N L, Dyer T E and Riis E 2018 Free-induction-decay magnetometer based on a microfabricated Cs vapor cell *Phys. Rev. Appl.* **10** 014002
[20] Wheeler M D, Newman S M, Orr-Ewing A J and Ashfold M N R 1998 Cavity ring-down spectroscopy *J. Chem. Soc., Faraday Trans.* **94** 337–51
[21] Romanini D, Kachanov A A, Sadeghi N and Stoeckel F 1997 CW cavity ring down spectroscopy *Chem. Phys. Lett.* **264** 316–22
[22] Berden G, Peeters R and Meijer G 2000 Cavity ring-down spectroscopy: experimental schemes and applications *Int. Rev. Phys. Chem.* **19** 565–607
[23] Berden G and Engeln R 2009 *Cavity Ring-Down Spectroscopy: Techniques and Applications* (New York: Wiley)
[24] Zhiyan Li *et al* 2019 Simultaneous measurement of NO and NO_2 by a dual-channel cavity ring-down spectroscopy technique *Atmos. Meas. Tech.* **12** 3223–36
[25] Müller T, Wiberg K B and Vaccaro P H 2000 Cavity ring-down polarimetry (CRDP): a new scheme for probing circular birefringence and circular dichroism in the gas phase *J. Phys. Chem.* A **104** 5959–68
[26] Müller T, Wiberg K B, Vaccaro P H, Cheeseman J R and Frisch M J 2002 Cavity ring-down polarimetry (CRDP): theoretical and experimental characterization *J. Opt. Soc. Am.* B **19** 125–41
[27] Sofikitis D, Bougas L, Katsoprinakis G E, Spiliotis A K, Loppinet B and Rakitzis T P 2014 Evanescent-wave and ambient chiral sensing by signal-reversing cavity ringdown polarimetry *Nature* **514** 76–9
[28] Bougas L, Sofikitis D, Katsoprinakis G E, Spiliotis A K, Tzallas P, Loppinet B and Peter Rakitzis T 2015 Chiral cavity ring down polarimetry: chirality and magnetometry measurements using signal reversals *J. Chem. Phys.* **143** 09B603_1
[29] Dupré P 2015 Birefringence-induced frequency beating in high-finesse cavities by continuous-wave cavity ring-down spectroscopy *Phys. Rev.* A **92** 053817
[30] Visschers J C, Tretiak O, Budker D and Bougas L 2020 Continuous-wave cavity ring-down polarimetry *J. Chem. Phys.* **152** 164202
[31] Bougas L, Katsoprinakis G E, Von Klitzing W, Sapirstein J and Rakitzis T P 2012 Cavity-enhanced parity-nonconserving optical rotation in metastable Xe and Hg *Phys. Rev. Lett.* **108** 210801
[32] Stamataki K, Papadakis V, Everest M A, Tzortzakis S, Loppinet B and Rakitzis T P 2013 Monitoring adsorption and sedimentation using evanescent-wave cavity ringdown ellipsometry *Appl. Opt.* **52** 1086–93
[33] Sofikitis D, Stamataki K, Everest M A, Papadakis V, Stehle J-L, Loppinet B and Rakitzis T P 2013 Sensitivity enhancement for evanescent-wave sensing using cavity-ring-down ellipsometry *Opt. Lett.* **38** 1224–6
[34] Sofikitis D *et al* 2015 Microsecond-resolved SDR-based cavity ring down ellipsometry *Appl. Opt.* **54** 5861–5
[35] Halmer D, Golo von B, Hering P and Mürtz M 2004 Fast exponential fitting algorithm for real-time instrumental use *Rev. Sci. Instrum.* **75** 2187–91
[36] Mazurenka M, Wada R, Shillings A J L, Butler T J A, Beames J M and Orr-Ewing A J 2005 Fast Fourier transform analysis in cavity ring-down spectroscopy: application to an optical detector for atmospheric NO_2 *Appl. Phys.* B **81** 135–41
[37] Everest M A and Atkinson D B 2008 Discrete sums for the rapid determination of exponential decay constants *Rev. Sci. Instrum.* **79** 023108
[38] Bostrom G, Atkinson D and Rice A 2015 The discrete fourier transform algorithm for determining decay constants—implementation using a field programmable gate array *Rev. Sci. Instrum.* **86** 043106
[39] Aboutanios E 2009 Estimation of the frequency and decay factor of a decaying exponential in noise *IEEE Trans. Signal Process.* **58** 501–9
[40] Aboutanios E 2011 Estimating the parameters of sinusoids and decaying sinusoids in noise *IEEE Instrum. Meas. Mag.* **14** 8–14
[41] Visschers J C, Wilson E, Conneely T, Mudrov A and Bougas L 2021 Rapid parameter determination of discrete damped sinusoidal oscillations *Opt. Express* **29** 6863–78
[42] van Veen F and Leijnen S 2020 The neural network zoo *Proceedings* **47** 9
[43] Bourlard He and Kamp Y 1988 Auto-association by multilayer perceptrons and singular value decomposition *Biol. Cybern.* **59** 291–4
[44] DeMers D and Cottrell G W 1993 Non-linear dimensionality reduction 5 *Advances in Neural Information Processing Systems* pp 580–7
[45] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7
[46] Theis L, Shi W, Cunningham A and Huszár F 2017 Lossy image compression with compressive autoencoders (arXiv:1703.00395)
[47] Lu X, Tsao Y, Matsuda S and Hori C 2013 Speech enhancement based on deep denoising autoencoder *Interspeech* vol 2013 pp 436–40
[48] Andrew N 2011 *CS294A Lect. notes* Stanford University
[49] Asperti A and Trentin M 2020 Balancing reconstruction error and Kullback-Leibler divergence in variational autoencoders *IEEE Access* **8** 199440–8
[50] Gagliardi G and Loock H-P 2014 *Cavity-Enhanced Spectroscopy and Sensing* vol 179 (Berlin: Springer)
[51] Boens Nel *et al* 2007 Fluorescence lifetime standards for time and frequency domain fluorescence spectroscopy *Anal. Chem.* **79** 2137–49

[52] Cundall R 2013 *Time-Resolved Fluorescence Spectroscopy in Biochemistry and Biology* vol 69 (Berlin, Heidelberg: Springer Science and Business Media)
[53] Spiliotis A K, Xygkis M, Klironomou E, Kardamaki E, Boulogiannis G K, Katsoprinakis G E, Sofikitis D and Rakitzis T P 2020 Optical activity of lysozyme in solution at 532 nm via signal-reversing cavity ring-down polarimetry *Chem. Phys. Lett.* **747** 137345
[54] Papadakis V, Everest M A, Stamataki K, Tzortzakis S, Loppinet B and Peter Rakitzis T 2011 Development of cavity ring-down ellipsometry with spectral and submicrosecond time resolution *Instrumentation, Metrology and Standards for Nanomanufacturing, Optics and Semiconductors V* vol 8105 (International Society for Optics and Photonics) p 81050L
[55] Yao Y-X and Pandit S M 1995 Cramér–Rao lower bounds for a damped sinusoidal process *IEEE Trans. Signal Process.* **43** 878–85
[56] Chollet Fçois *et al* 2015 Keras (available at: https://keras.io)
[57] Wahl E H, Tan S M, Koulikov S, Kharlamov B, Rella C R, Crosson E R, Biswell D and Paldus B A 2006 Ultra-sensitive ethylene post-harvest monitor based on cavity ring-down spectroscopy *Opt. Express* **14** 1673–84