

Direct ultrashort-pulse intensity and phase retrieval by frequency-resolved optical gating and a computational neural network

Marco A. Krumbügel, Celso L. Ladera, Kenneth W. DeLong, David N. Fittinghoff, John N. Sweetser, and Rick Trebino

Sandia National Laboratories, MS 9057, Livermore, California 94551-0969

Received October 6, 1995

Ultrashort-laser-pulse retrieval in frequency-resolved optical gating has previously required an iterative algorithm. Here, however, we show that a computational neural network can *directly* and rapidly recover the intensity and phase of a pulse. © 1996 Optical Society of America

Recently several techniques have been developed for measuring the full time-dependent intensity and phase of ultrashort laser pulses.¹⁻³ Although these techniques represent a vast improvement in our ability to characterize such pulses, they require additional effort, both in the apparatus and in the extraction of the pulse intensity and phase from the experimental trace. Frequency-resolved optical gating^{3,4} (FROG) is perhaps the simplest of these methods to implement experimentally, but pulse extraction in FROG requires an iterative algorithm, which generally takes a minute or so to converge. As FROG is also to our knowledge the only such method to have demonstrated single-shot operation, it is important that pulse extraction in FROG be performed more rapidly. The purpose of this Letter is to demonstrate that pulse extraction in FROG can, in fact, be performed directly and rapidly by a computational neural network.

In FROG, a nonlinear-optical autocorrelation signal is frequency resolved by a spectrometer to produce a FROG trace, which is a spectrogram of the complex pulse field versus time, $E(t)$. The FROG trace visually displays the pulse frequency versus time. It also uniquely determines $E(t)$.³ In its polarization-gate version, the FROG trace is given by

$$I_{\text{FROG}}(\omega, \tau) = \left| \int_{-\infty}^{\infty} E(t)|E(t - \tau)|^2 \exp(-i\omega t) dt \right|^2, \quad (1)$$

where ω is the frequency and τ is the delay between the two replicas of the pulse used to make the trace.

Equation (1) can be thought of as a function that maps the pulse field to its FROG trace. The goal of the pulse-extraction problem is then to invert this function to produce the pulse field in terms of its FROG trace. Unfortunately, no closed-form solution exists for this inverse mapping, so an iterative phase-retrieval algorithm is used to find the pulse field for a given trace.³ This algorithm works well and generally converges in a minute or so, but, for many applications more-rapid retrieval is quite important.

A powerful tool for simulating difficult to compute complex functions is the computational neural network, or the neural net.⁵⁻⁷ It has been shown that neural

nets can accurately simulate any function.^{6,7} Success has been attained in such diverse applications as handwriting recognition, high-energy physics,⁸ and stock-market prediction. In its simplest form a neural net (see Fig. 1) consists of several layers of interconnected nodes, each of which performs a series of multiplications, additions, and nonlinear filtering operations. The input data comprise the input nodes. At each node of a second layer of nodes, the hidden layer, the value is the sum of the products of the input nodes and a set of weights to be determined. These values are then input into filter functions, typically a hyperbolic tangent, to introduce some nonlinearity. This process is then repeated at a third set of nodes, the output-layer nodes, which comprise the solution. The net is trained, that is, the weights are determined, by generating a set of known input and output training values and iteratively minimizing the error between the known output values and the net's output values.

Once a neural net is trained—generally a time-consuming process—its use in practice is direct and essentially instantaneous, typically requiring less than a second on any computer. Neural nets are also known for their insensitivity to noise. Finally, they seem to operate best when the human mind is also adept at determining the solution. All these characteristics imply that a neural net should be ideal for pulse retrieval in FROG.

In this Letter we show that computational neural networks can directly obtain the intensity and phase of a pulse from its FROG trace. Our demonstration involved a limited set of theoretical pulses and was limited in its generality by the net training time. It is

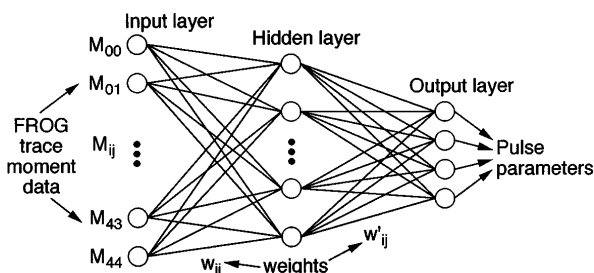


Fig. 1. Neural network for retrieval of ultrashort laser pulses from FROG traces.

Table 1. Ranges of Values for the Four Parameters Used in the Training of the Neural Net^a

	$\Delta\omega$	β	γ	δ
Range	6–18	-1.0–+1.0	-1.0–+1.0	-0.5–+0.5
Training error	5.8%	4.6%	4.6%	4.5%
Validation error	4.2%	4.5%	4.0%	6.1%

^aThe second row shows the rms error for each parameter achieved by the net for the training set of pulses after convergence was declared; the third row shows the corresponding rms errors for the validation set of pulses after training.

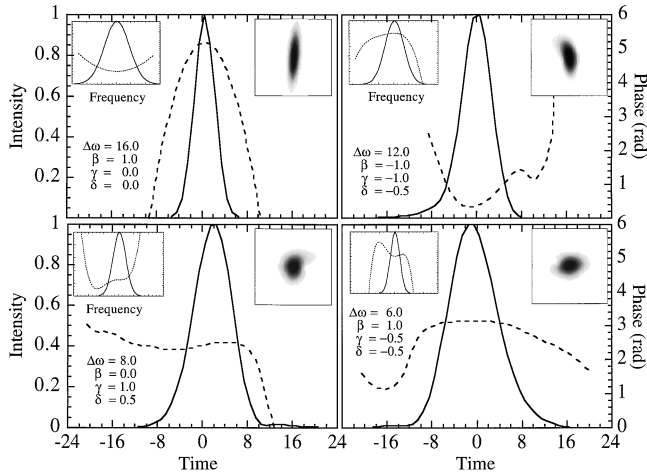


Fig. 2. Intensity (solid curves) and phase (dashed curves) of four pulses typical of those used to train the neural net. The insets in the upper left show the pulses versus frequency; the insets in the upper right show the corresponding FROG traces (frequency versus time).

thus only a proof of this principle and not a complete solution. However, because fast and powerful parallel-processing hardware is now available, waveform recovery for arbitrary pulses appears likely. It should be emphasized that such hardware is needed only for training the net and is not necessary for using the net.

We used a set of pulses whose complex frequency-domain field, $E(\omega)$, is parameterized by four parameters: the spectral width ($\Delta\omega$) and the quadratic (β), cubic (γ), and quartic (δ) spectral phase coefficients:

$$E(\omega) = \exp\left[-2 \ln 2 \frac{(\omega - \omega_0)^2}{\Delta\omega^2}\right] \exp\left[i\beta\left(\frac{\omega - \omega_0}{\Delta\omega}\right)^2 + i\gamma\left(\frac{\omega - \omega_0}{\Delta\omega}\right)^3 + i\delta\left(\frac{\omega - \omega_0}{\Delta\omega}\right)^4\right]. \quad (2)$$

To train the network we used a set of FROG traces generated by choosing 300 pulses, that is, 300 sets of the above four parameters. The parameter ranges are shown in Table 1 and were limited solely by the requirement that the resulting trace fit within a 128 × 128 grid. Figure 2 shows examples of these pulses.

A FROG trace typically contains 64 × 64 or more data points—far too much input data for a neural net. Indeed, there is much redundancy in such a trace, as it determines at most 64 intensity points and 64 phase points of the corresponding electric field. To reduce the required number of nodes, we performed feature extraction on the trace to extract only the essential information. Our implementation of feature

extraction involved computing the 25 lowest-order two-dimensional integral moments⁹ of the FROG traces:

$$M_{pq} \equiv \frac{1}{N^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I_{\text{FROG}}(\omega, \tau) \omega^p \tau^q d\omega d\tau, \quad (3)$$

with $p, q = 0, 1, 2, 3, 4$. Moments can be shown to determine completely the trace if N^2 of them are used. The lowest-order moments yield basic features of the trace, such as its area and symmetries. The use of moments results in significant compression of the input data. An additional advantage is that a trace of any size may be used as input.

The structure of our net is straightforward. Twenty-five input-layer nodes take the values of the moments, and four output-layer nodes yield the pulse parameters. The number of hidden-layer nodes, however, must be determined. Use of too few hidden-layer nodes can fail to model the full complexity of the pulse extraction function and yield high errors. Use of too many hidden-layer nodes, however, can cause overtraining (accurately yielding all pulses in the training set but failing to reproduce new pulses later). We have found that 10 hidden-layer nodes work best. The symmetric sigmoid function $\tanh(x)/2$ was used as a filter at each hidden- and output-layer node, and all parameter values were linearly scaled to

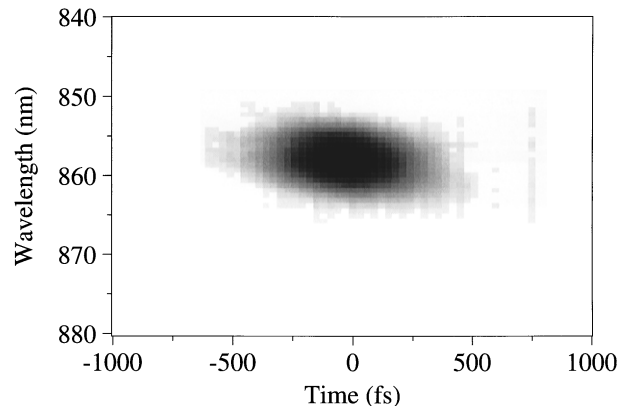


Fig. 3. Measured FROG trace.

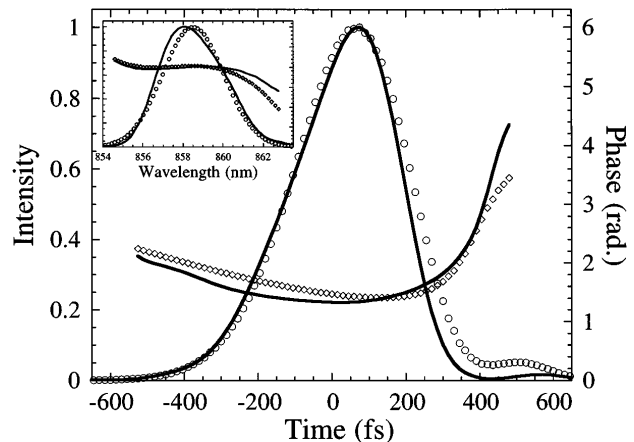


Fig. 4. Intensity and phase of the trace shown in Fig. 3, retrieved by the neural net (circles and diamonds) and the iterative FROG algorithm (solid curves) in the time domain and in the frequency domain (inset).

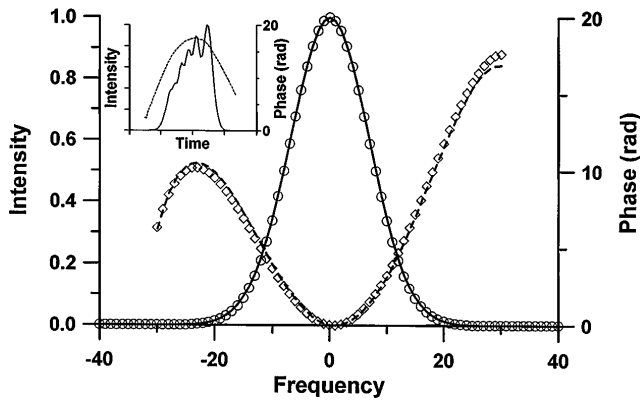


Fig. 5. Intensity and phase of a typical test pulse (solid and dashed curves) and those reconstructed by the neural network in the presence of 5% noise (circles and diamonds).

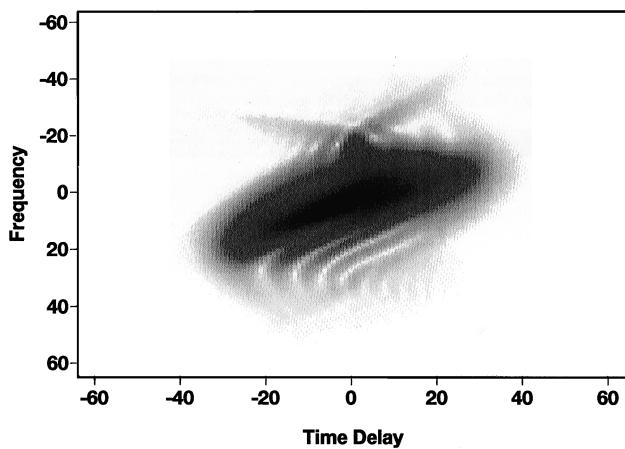


Fig. 6. FROG trace of the pulse shown in Fig. 5. The actual and retrieved traces are visually identical, so only one is shown.

the interval $[-0.4, 0.4]$ to best utilize the $(-0.5, 0.5)$ range of this function.

We used the Quickprop algorithm¹⁰ running on a Silicon Graphics Indy workstation to train the net. One pass of the training set across the Quickprop algorithm is called an epoch; our training involved 20,000 epochs. Convergence was declared when the rms error between the actual and the retrieved pulse parameters was below 5% and occurred in 6 h.

After training, we tested the network performance with a validation set of new pulses not in the training set and chosen randomly within the ranges given in Table 1. Specifically, this involved using the trained network to estimate the parameters of these new pulses from their FROG trace moments. Invariably, the net produced very good estimates of all four parameters of each pulse in the test set (Table 1). These errors are slightly smaller than the errors obtained for the training set because the pulses of the validation set are randomly chosen and therefore emphasize the center of the four-dimensional parameter space, whereas the pulses of the training set emphasize its perimeter.

We also tested the network performance on the experimental FROG trace shown in Fig. 3. A

comparison between the intensity and the phase predicted by the neural network and the intensity and the phase retrieved by the iterative algorithm is shown in Fig. 4. Although the experimental pulse is not of true Gaussian shape in the frequency domain and therefore cannot have been accurately described by only one output parameter $\Delta\omega$, the intensity and the phase of the pulse are fairly well reproduced by the neural net in the time domain as well as in the frequency domain.

In an additional investigation running on a Macintosh computer, we tested a net of the same configuration in the presence of multiplicative noise. Using a commercial MATLAB feed-forward backpropagation algorithm,⁵ hyperbolic-tangent filter functions, and a training set of 144 pulses, we found that adding 5% multiplicative noise to the FROG traces typically increases the error between the recovered and actual pulse parameters by a factor of 2. Figure 5 shows a retrieved pulse for a typical case of this investigation, using a noise-contaminated trace. In Fig. 6 we show the FROG trace of the pulse in Fig. 5.

We have used only polarization-gate FROG traces in this study, but the generality of neural nets will allow any type of FROG trace or other pulse measure to be used. For example, FROG traces made with noninstantaneously responding nonlinear-optical media have been shown to be retrievable with an iterative algorithm, but the process is extremely slow.¹¹ A neural network will perform this task quickly.

The authors acknowledge the support of the U.S. Department of Energy, Basic Energy Sciences, Division of Chemical Sciences, and a Cooperative Research and Development Agreement. C. L. Ladera acknowledges the support of the Universidad Simon Bolivar and of Fundayacuco, Venezuela. M. A. Krumbügel thanks Anthony E. Siegman and the Alexander von Humboldt Foundation for support.

References

1. J.-C. M. Diels, J. J. Fontain, I. C. McMichael, and F. Simoni, *Appl. Opt.* **24**, 1270 (1985).
2. J. L. A. Chilla and O. E. Martinez, *Opt. Lett.* **16**, 39 (1991).
3. R. Trebino and D. J. Kane, *J. Opt. Soc. Am. A* **10**, 1101 (1993).
4. K. W. DeLong, R. Trebino, and D. J. Kane, *J. Opt. Soc. Am. B* **11**, 1595 (1994).
5. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing* (MIT Press, Cambridge, Mass., 1986), Vols. 1 and 2.
6. R. Hecht-Nielsen, in *Proceedings of International Conference on Neural Networks III* (Institute of Electrical and Electronics Engineers, New York, 1987), p. 11.
7. V. Kurková, *Neural Networks* **5**, 501 (1992).
8. A. L. Perrone, P. Castiglione, G. Basti, and R. Messi, *Proc. Soc. Photo-Opt. Instrum. Eng.* **2243**, 540 (1994).
9. M. R. Teague, *J. Opt. Soc. Am.* **70**, 920 (1980).
10. S. E. Fahlman, "Faster-learning variations on backpropagation: an empirical study," in *Proceedings of the 1988 Connectionist Models Summer School* (Morgan Kaufman, San Mateo, Calif., 1988), p. 38.
11. K. W. DeLong, C. L. Ladera, and R. Trebino, *Opt. Lett.* **20**, 486 (1995).