



OPEN

Serial and parallel convolutional neural network schemes for NFDN signals

Wen Qi Zhang^{1✉}, Terence H. Chan² & Shahraam Afshar Vahid¹

Two conceptual convolutional neural network (CNN) schemes are proposed, developed and analysed for directly decoding nonlinear frequency division multiplexing (NFDN) signals with hardware implementation taken into consideration. A serial network scheme with a small network size is designed for small user applications, and a parallel network scheme with high speed is designed for places such as data centres. The work aimed at showing the potential of using CNN for practical NFDN-based fibre optic communication. In the numerical demonstrations, the serial network only occupies 0.5 MB of memory space while the parallel network occupies 128 MB of memory but allows parallel computing. Both network schemes were trained with simulated data and reached more than 99.9% accuracy.

Nonlinear Fourier Transform (NFT) has been proposed as an alternative technique for fibre optics communication to break through the Shannon linear capacity limit due to the nonlinearity of optical fibres^{1–4}. It has already been experimentally demonstrated that NFT can outperform conventional linear Fourier transform-based systems^{5–7}. However, NFT-based communication methods have not yet reached the maturity level of conventional methods. One reason for this is the complexity of NFT calculations, where speed and accuracy are the two main bottlenecks for practical applications^{8,9}. Despite the development of fast NFT algorithms^{10–13} which has improved the speed of NFT calculation, the lack of hardware implementations of NFT due to algorithm complexity remains a challenge. Very recently, a conceptual hardware implementation has been proposed on an FPGA platform¹⁴. However, due to the complexity of NFT algorithms, implementing and optimizing hardware designs require a lot of effort before any products can be built for practical use and with perform better than existing conventional devices.

In recent years, machine learning (ML) has demonstrated great success in many areas of science and technologies. Although the development of ML for optical communication is still in its early stages, ML has shown potentials for solving many challenges^{15–21}, including the field of NFT-based communication^{22–29}. At the same time, due to the popularity of machine learning, hardware implementation of deep neural networks has become an active research area with supports from mega-corporations such as Google and Nvidia^{30–36}. Hence, there is an opportunity to take advantage of the rapid development of deep learning hardware and use it for NFT applications.

Early work of using ML for NFT-based communication was mainly focused on post-processing, such as constellation classification²⁵ and equalisation^{23,26}, instead of replacing the direct NFT calculation. An attempt of using a neural network to decode nonlinear QPSK modulation was carried out by Jones et al.²², but the development of the network was limited only to second-order soliton pulses. Recently, a convolutional neural network (CNN) was proposed to directly decode nonlinear frequency division multiplexing (NFDN) signals with complexity (16-QAM with 128 subcarriers) which was comparable to experimental setups²⁹. However, in this previous work, the network was not used efficiently, especially for signals with many subcarriers, since for every subcarrier, a separate network was needed. Furthermore, the previous work only demonstrated networks for a fixed signal power without considering fibre loss and in the case of lump amplification. While this paper was being reviewed, two new articles were published^{27,28}. Both of them were aimed at obtaining the nonlinear spectrum of a signal. In the first article²⁷, two CNN sub-structures were applied to process the real and imaginary parts of the signal in parallel. However, the training data used in this work only has 15 subcarriers and four possible choices for each subcarrier. Therefore, the scalability of this proposed network is also unknown. Furthermore, the training data does not include variation in signal propagation distance, absorption loss and variation in power, hence

¹Laser Physics and Photonic Devices Laboratories, STEM, University of South Australia, Adelaide, Australia. ²Institute for Telecommunications Research, University of South Australia, Adelaide, Australia. ✉email: wenqi.zhang@unisa.edu.au

the generality is hard to evaluate. In the second work²⁸, a network architecture similar to the one in the work²⁹ is used for both direct and inverse NFT. However, this article did not give any details on the training data. Hence, the generality and scalability of their network cannot be evaluated.

In this paper, we develop two network design ideas with hardware implementation kept in mind; a serial scheme aimed at a small network size and a parallel scheme aimed at high speed. In each scheme, a custom transformation layer is designed using the properties of NFT that enables the network to work with signals at different propagation lengths and power levels. An optical pulse propagation model is used to simulate signal propagation in optical fibre including the effects of fibre nonlinearity, chromatic dispersion, absorption loss, Raman scattering and lump amplification to closely mimic realistic experimental settings. Using the simulation, we generated 128-subcarrier 16-QAM signals with variable propagation length and power level for network training and validation.

The paper is divided into six sections apart from the Introduction. In the second section, a quick introduction to nonlinear Fourier transform and nonlinear frequency division multiplexing is given. In the next section, the basic parameters and descriptions are given on the generation of the training and validation data. The serial and parallel network schemes are discussed in the following sections. Finally, a conclusion and a method section are given at the end.

Modulation of continuous nonlinear spectrum

Transforming a temporal signal $q(t)$ into the nonlinear Fourier domain is achieved by solving the following differential equation¹

$$v_t = \begin{pmatrix} -i\lambda & q(t) \\ -q^*(t) & i\lambda \end{pmatrix} v, \quad (1)$$

with the initial condition:

$$v(t, \lambda) = \begin{pmatrix} v_1(t, \lambda) \\ v_2(t, \lambda) \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} e^{-j\lambda t}, \quad t \rightarrow -\infty \quad (2)$$

where v_t is the temporal derivative of v and λ is the nonlinear frequency. Two time invariant coefficients $a(\lambda)$ and $b(\lambda)$ can be found using the following limits³:

$$a(\lambda) = \lim_{t \rightarrow \infty} v_1(t, \lambda) e^{j\lambda t}, \quad b(\lambda) = \lim_{t \rightarrow \infty} v_2(t, \lambda) e^{-j\lambda t}. \quad (3)$$

The nonlinear spectrum of signal $q(t)$ is thus defined as

$$Q(\lambda) = \begin{cases} b(\lambda)/a(\lambda) & \text{for } \lambda \in \mathbb{R}, \\ b(\lambda)/a'(\lambda) & \text{for } \lambda \in \mathbb{C}^+ \text{ and } a(\lambda) = 0, \end{cases} \quad (4)$$

where $a'(\lambda) = \frac{\partial a(\lambda)}{\partial \lambda}$. The nonlinear Fourier spectrum is continuous for $\lambda \in \mathbb{R}$ and discrete when $a(\lambda) = 0$ for $\lambda \in \mathbb{C}^+$. The spectrum in both continuous and discrete region can be used to carry information, but in this work, we focus on the continuous part only.

Nonlinear frequency division multiplexing (NFDm) is a multiplexing technique borrowed from orthogonal frequency division multiplexing (OFDM) in conventional fibre optic communication, in which the linear spectrum of OFDM is replaced by a nonlinear spectrum³ and becoming popular in recent years^{6,37–41}. The modulation is usually applied to the continuous part of the nonlinear spectrum $Q(\lambda)$ directly, but modulating $b(\lambda)$ (b-modulation) has also been used as it results in pulses with well defined temporal windows^{7,42–46}.

The modulation can often be expressed as

$$s(\lambda) = \sum_n c_n w_n(\lambda), \quad (5)$$

where n is the number of subcarriers, c_n is the complex data symbols, $w_n(\lambda)$ is the carrier wave. Conventionally, sinc functions are used as $w_n(\lambda)$ ³⁷. Other choices of functions such as raised cosine or a flat-top window with cardinal sine carrier waveform have also been proposed⁴³. The function $s(\lambda)$ is the modulated spectrum. It is often scaled to match the desired signal power. For Q-modulation case, $Q(\lambda) = s(\lambda)$. An inverse NFT can be performed to obtain the temporal signal $q(t)$ ³⁷. In the case of b-modulation, $b(\lambda) = s(\lambda)$, and an additional step is needed to generate $a(\lambda)$ from $b(\lambda)$ before the inverse NFT step can be taken⁴².

Inspired by the work⁴³, we used the following carrier wave functions for this work.

$$w_n(t) = \text{sinc}\left(4\frac{t}{T_0}\right) \cdot \exp\left[-2\left(\frac{t}{\tau T_0}\right)^2 - in\pi \cdot 8\frac{t}{T_0}\beta\right] \quad (6)$$

$$= \text{sinc}\left(4\frac{t}{T_0}\right) \cdot \exp\left[-2\frac{t}{T_0}\left(\frac{t}{\tau^2 T_0} + i4n\pi\beta\right)\right], \quad (7)$$

where T_0 denotes the width to the 4th zero of the sinc function [note: here $\text{sinc}(t) = \sin(\pi t)/(\pi t)$], τ is a free parameter that adjusts the width of the Gaussian, which also related to the sharpness of the flat-top waveform in the frequency domain, and β adjusts the separation between subcarriers.

In the nonlinear frequency domain, the corresponding $w_n(\lambda)$ can be obtained by replacing the angular frequency in $w_n(\omega)$ with 2λ (or -2λ depending on the choice of the sign in the Fourier transform), where $w_n(\omega)$ is the linear Fourier transform of $w_n(t)$;

$$w_n(\lambda) = \frac{T_0}{8\sqrt{2\pi}} \left\{ \text{erf} \left[\frac{\tau T_0}{\sqrt{2}} \left(\lambda - \frac{4\pi\beta}{T_0} n + \frac{2\pi}{T_0} \right) \right] - \text{erf} \left[\frac{\tau T_0}{\sqrt{2}} \left(\lambda - \frac{4\pi\beta}{T_0} n - \frac{2\pi}{T_0} \right) \right] \right\}. \quad (8)$$

From Eq. (8), one can see the nonlinear frequency separation between subcarriers is $4\pi\beta/T_0$, whilst $4\pi/T_0$ is approximately the bandwidth of each subcarrier. For the rest of the work, $\tau = 1$ and $\beta = 1$ are used.

Generation of training data

The preparation of the training data is crucial to the performance of a neural network. Ideally, the training data should be generated experimentally such that the network can be tuned to the specific experimental setup. However, for demonstration purposes, we generate the training data through numerical simulation while the simulation conditions are tuned to match those of the experiments closely. Assuming a signal generator with a sampling rate of 96 GS/s with total electronic 3-dB bandwidth of 28 GHz. A time window of 21.33 ns is used for each burst of the signal. The laser source has a line-width of 100 Hz at the central wavelength of 1550 nm. Standard single-mode fibres are used with the propagation loss of 0.2 dB/km, group velocity dispersion of $-21.7 \text{ ps}^2/\text{km}$ ($\beta_3 = 18.6 \text{ ps}^3/\text{km}$) and nonlinearity of $1.1 \text{ W}^{-1} \text{ km}^{-1}$. A maximum of 10 optical fibre spans with 50 km per span are used in the simulation with point amplification at the end of each span. For each burst of signal, a random number of spans between 1 and 10 is chosen. Lump amplification is applied in the simulation and a path-average model is used to counteract the effects of fibre loss and lump amplification⁴⁷. A quadrature amplitude modulation is used (16-QAM) for encoding subcarriers with direct mapping of numbers 0–15 to the symbols. The numbers are then used as labels for the network training algorithm.

We desire to have the network perform equally at different input power levels in the range of -20 to 0 dBm, hence we need a way to randomly generate the training data with power evenly distributed within this range to avoid bias of the final network towards certain input power. To achieve this, we have developed and used an approximated energy calibration model, which is discussed in the next section. The noise background in the training data is chosen to be -40 dBm per amplification.

An approximated model for energy calibration. The energy of a burst can be evaluated using its nonlinear spectrum $Q(\lambda)$ as

$$E = \frac{1}{\pi} \int \ln(1 + |A \cdot Q(\lambda)|^2) d\lambda, \quad (9)$$

where A is a real scaling factor through which different burst energy can be set. The energy E does not scale linearly with A . Depending on the choice of the carrier wave and the number of subcarriers, the same A results in different energy levels. This is particularly difficult for generating training data since a uniform distribution of power on the log scale is needed (-20 to 0 dBm in this work). Here we present a fitting model that gives an approximated mapping between A and the average burst energy.

Firstly, we discretize the integral in Eq. (9) as

$$E = \frac{1}{\pi} \sum_i \ln(1 + A^2 |Q_i|^2) \Delta\lambda \quad (10)$$

where i changes from 1 to the number of discrete points across the nonlinear spectrum. Rearranged the equation, we have

$$E \frac{\pi}{\Delta\lambda} = \sum_i \ln(1 + A^2 |Q_i|^2) \quad (11)$$

$$= \ln \left[\prod_i (1 + A^2 |Q_i|^2) \right]. \quad (12)$$

Let $C = \prod_i (1 + A^2 |Q_i|^2)$ and take natural logarithm on both sides of the equation above, we have

$$\ln E + \ln \frac{\pi}{\Delta\lambda} = \ln \ln C. \quad (13)$$

Next, we normalize the symbol constellation to its mean amplitude, $c_n/|c_n|$, and the carrier wave function to its peak amplitude, $w_n(\lambda)/\max(w_n(\lambda))$. From now on, all $Q(\lambda)$ shall be generated using the normalized constellation and carrier waves. After that, we assume a new constant $|Q|$ such that

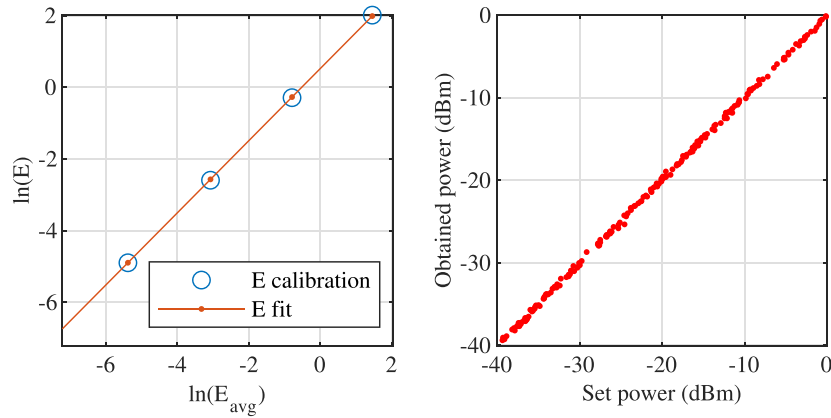


Figure 1. Power calibration model. Left: linear fitting of the model. Right: 200 randomly generated pulses using the energy calibration model.

$$E \frac{\pi}{\Delta\lambda} = \ln \left[\prod_i (1 + A^2 |Q_i|^2) \right] \tag{14}$$

$$= \ln \left(1 + A^2 |\bar{Q}|^2 \right)^n \tag{15}$$

$$E \frac{\pi}{\Lambda} = \ln \left(1 + A^2 |\bar{Q}|^2 \right), \tag{16}$$

where n is the number of samples in λ and $\Lambda = n\Delta\lambda$ is the window size in the nonlinear spectrum. Now, we define a new parameter A' such that

$$A'|1| = A|\bar{Q}| = \sqrt{e^{\left(\frac{\pi}{\Lambda} E\right)} - 1}. \tag{17}$$

And we choose a few points E' within the energy range $[E_{\min}, E_{\max}]$ to calculate A' using Eq. (17). Once, we get A' , we replace A in Eq. (9) with A' and calculate the actual averaged pulse energy E_{avg} using randomly generated $Q(\lambda)$. Depending on the choice of carrier wave function, QAM format, number of subcarriers, etc., E_{avg} differs from E' Figure 1 shows an example.

Finally, a least square fit is applied to $\ln E_{\text{avg}}$ and $\ln E'$,

$$\begin{pmatrix} p_1 \\ p_0 \end{pmatrix} = \text{pinv}([\ln E_{\text{avg}} \ J]) \times \ln E', \tag{18}$$

where $\ln E_{\text{avg}}$ and $\ln E'$ are column vectors, J is an all-ones column vector and pinv is the matrix pseudo-inverse function. And finally the calibrated A_{cal} is calculated as

$$A_{\text{cal}} = \sqrt{e^{\frac{\pi}{\Lambda} (p_1 \ln E + p_0)} - 1}.$$

The serial CNN scheme

In the original work²⁹, to decode each subcarrier, a transformation is needed to be performed on the received signal to remove the propagation induced temporal shift and phase change. Then, each subcarrier is decoded using its corresponding network. This process is inefficient since the differences between the nonlinear spectra of subcarriers are rather small, except for the ones on the edges of the nonlinear spectrum. There are overlaps between the networks from previous work and in principle, all of them can be combined into one. Furthermore, during our study, we realised that the evolution of the linear spectrum of subcarrier k is mostly affected by its neighbouring subcarriers. Therefore, the input to the network can be reduced significantly by applying a gate function to the input pulse in the linear frequency domain, hence greatly reduce the parameter space of the network.

Figure 2 illustrates the idea. An example pulse in the linear frequency domain is aligned with 128 subcarrier waves $w_n(\lambda)$, where the nonlinear frequency is half of the linear frequency ($2\lambda = \omega$). For any arbitrary subcarrier (between two black vertical lines), one adjacent subcarrier on each side (between two red lines) is used as the input to the network.

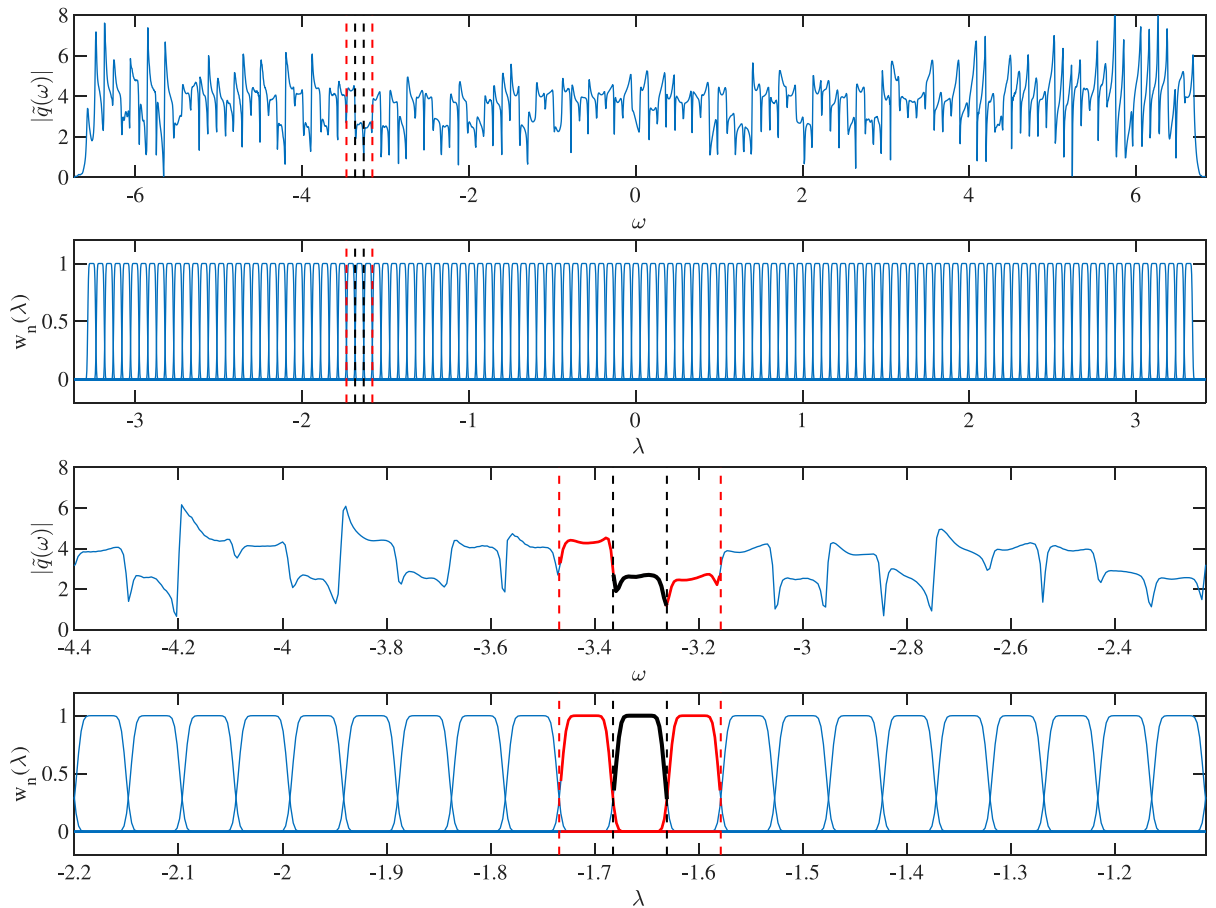


Figure 2. An example signal is shown in the linear frequency domain compared with subcarriers waves in the nonlinear frequency domain (top two figures). Two red vertical lines indicate the gated pulse that is used for decoding. The two black vertical lines indicate the subcarrier to be decoded by the network. Bottom two figures are the zoomed in view of the top figures, respectively, for clarity.

With these ideas in mind, we design a serial network scheme that works with all subcarriers. A network is trained for only 1 subcarrier (subcarrier $n = 0$). To decode any other subcarriers using this network, a segment of the pulse spectrum $q(\omega)$ around a subcarrier (between the two red lines) is taken out and shifted using Eq. (19);

$$\tilde{q}_k(\omega, z) = \tilde{q}(\omega - 2\lambda_k, z)e^{jt_k(\omega - \lambda_k)}, \tag{19}$$

where λ_k is the central nonlinear frequency of subcarrier k , $t_k = 4\lambda_k z$. This equation shifts the central frequency of subcarrier k to subcarrier 0 and removes the phase change due to propagation from the nonlinear spectrum²⁹. Note that a negative sign is used in the Fourier transforms when getting $\tilde{q}(\omega)$ from $q(t)$. The shift of Eq. (19) is built into the network as a transform layer.

All the subcarrier segments are collected in a queue and feed into the network. Figure 3 illustrates the idea. For the first and last subcarrier, zero paddings are applied. The output of the network is also a queue in the same order as the input. For each subcarrier input, the network outputs the decoded data corresponding to the mapping of the symbol c_n . In this work, it is number 0–15.

The schematic of the CNN is shown in Fig. 4. In the figure, only three convolutional layers are shown. However, depending on the input signal, e.g. the carrier wave function and the width of the spectrum segment, the network depth can be adjusted to maximize the network accuracy.

The network used in this work contains 4 convolutional layers. The input to the network is a 32-rows long by 2-columns wide array with each column corresponding to the real and imaginary parts of the 32 sample points of the input segment. Equation (19) is implemented into the network as a transform layer. All the convolutional layers have 64 kernels with a size of 3×3 for the first layers and 3×1 for the rest of the layers.

The network is trained using the Adam optimizer⁴⁸ with L2 regularization. 10,000 simulated pulses were generated within which 99% is used for training and 1% is used for validation. Since each pulse is sliced into 128 segments, the size of validation dataset is 12,800 samples. The decay rate of gradient and squared gradient moving averages of the Adam optimizer are set to be 0.9 and 0.99, respectively. The L2 regularization factor is set to 0.00002. Cross-entropy is used in the training algorithm as the loss function⁴⁹. As shown in Fig. 5, the network converges rather quickly. The accuracy of the network, defined as the percentage of correctly predicted symbols in the total number of symbols, reaches 90% in less than a single epoch and reaches over 99% before epoch 5. The

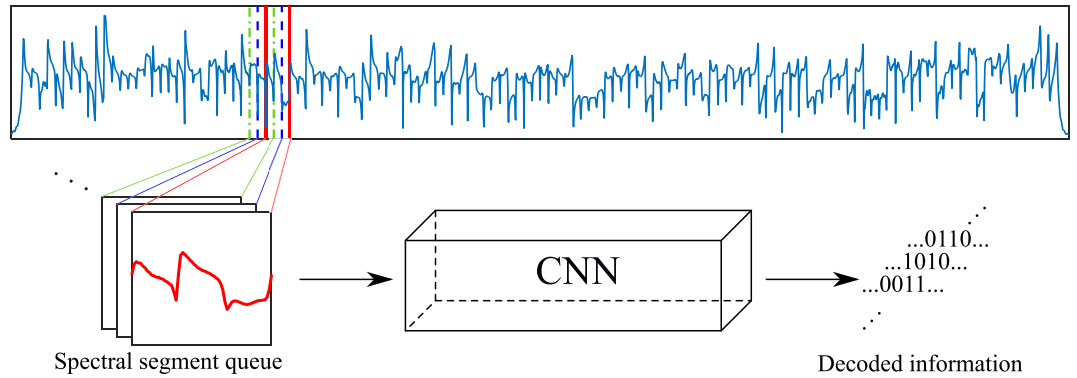


Figure 3. Illustration of the serial network decoding scheme. Segments of the spectrum corresponding to every subcarrier are queued up and passed through the network in serial. The spectra between the red solid, blue dashed and green dash-dotted lines are the segments of adjacent subcarriers in a queue.

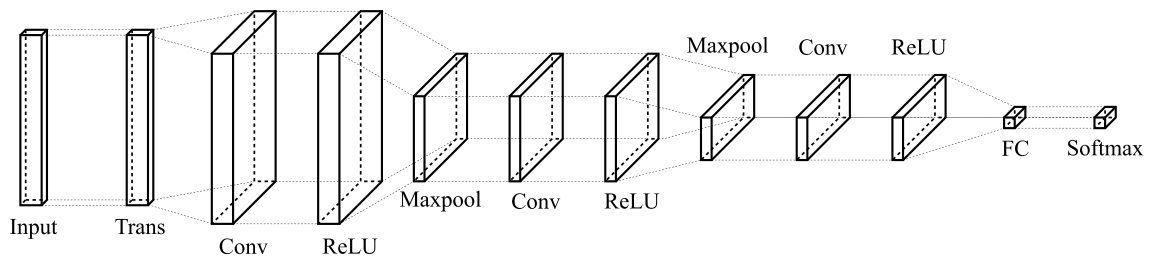


Figure 4. The conceptual schematic of the serial network. *Trans* transform layer, *Conv* convolutional layer, *ReLU* rectified linear unit activation layer, *Maxpool* max-pooling layer, *FC* fully connected layer, *Softmax* softmax layer. The number of convolution, activation and pooling layers can be added or removed depending on the input signal.

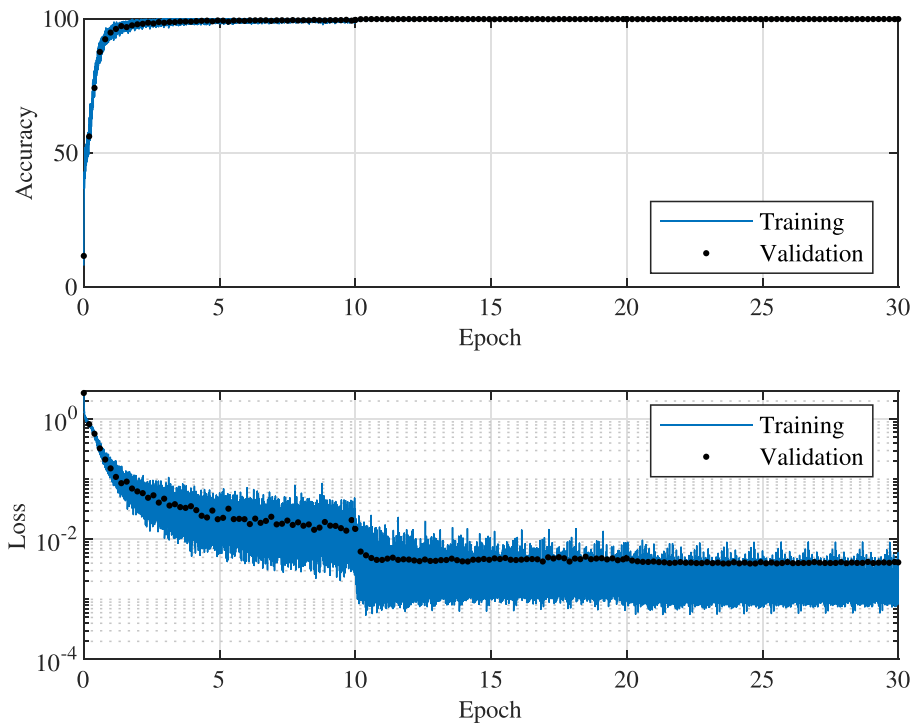


Figure 5. The accuracy and loss as functions of epochs for the serial network scheme. Learning rate changes take place at epoch 11 and 21.

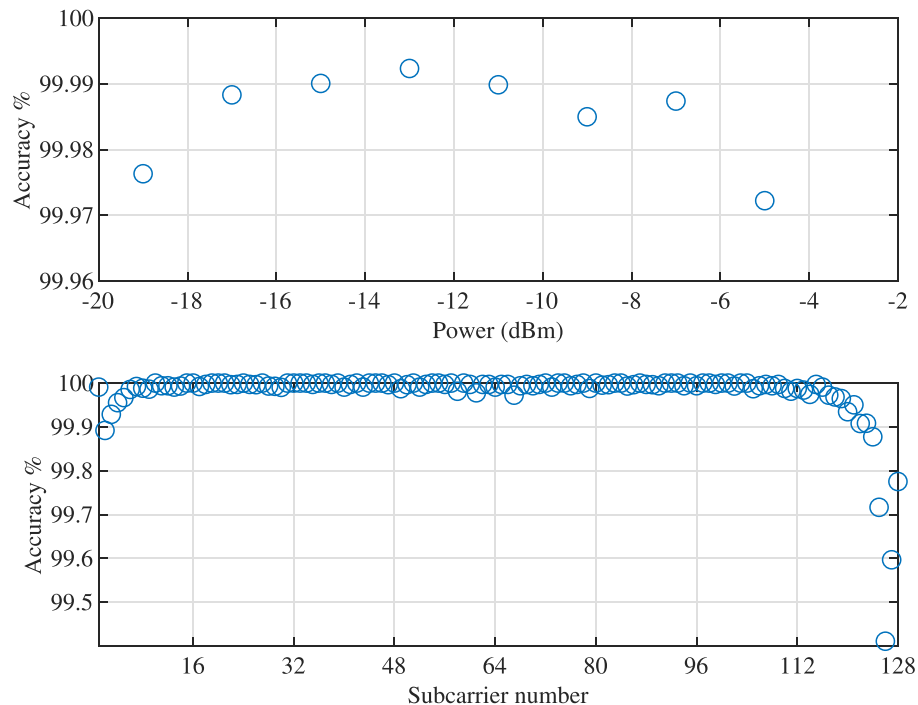


Figure 6. The accuracy for different pulse power and individual subcarrier.

learning rate for the first 10 epochs is set to 0.002 and then reduced to 0.0002 for the next 10 epochs as the convergence has slowed down significantly at epoch 10. A clear jump in the loss plot reflects the reduction of learning rate. The accuracy approaches 99.9% after the reduction. The learning rate was further reduced to 0.00002 for the third 10-epoch period but no further accuracy improvement was observed. From the loss value, we notice the step-wise drops at epoch 10 and 20 are corresponding to the change of learning rate. The loss and accuracy saturate after 21 epochs while the training loss is slightly lower than validation loss which indicates the network may have reached its capability limit for further improvements with the available training data. As an indication, we show the signal features that have been learnt by each convolutional layer in the Supplementary Material.

Looking at the network's prediction accuracy for different pulse power and individual subcarriers as shown in Fig. 6, we notice that the accuracy remains relatively constant for all ranges of pulse power. This can be the result of the fact that the range of power are covered by the training data so that the network is tuned to balance the loss values for different powers. For individual subcarriers, the accuracy is lower on the two edges of the spectrum (subcarrier 2–5 and 117–128), where in the middle, the accuracy is the same and at a high level. Interestingly, subcarrier 1 is on the very edge of the spectrum but still has high accuracy which indicates slight bias in the network toward the first subcarrier. The cause of this phenomenon needs further investigation.

This network scheme optimizes the size of the parameter space. It allows hardware implementation of the network on relatively small chips (the example network only occupies 0.5 MB of memory). Although every subcarrier has to be decoded sequentially, a pipe-lined hardware design is suitable for this network to minimize the speed penalty.

The parallel CNN scheme

The serial scheme is aimed at small network size, which allows the use of cheap hardware for network implementation with a small speed penalty making it suitable for small end users. But for big end-users, such as data centres, high performance is the major concern instead of hardware costs. Hence, here we propose a parallel network scheme where all subcarriers can be decoded at once.

In the serial scheme, every subcarrier is taken out of the linear spectrum of the signal with its neighbouring subcarriers before being sent into the network. To decode all subcarriers, the spectral data of each subcarrier is used multiple times in the subsequent decoding process, which can be saved if all the subcarriers can be decoded simultaneously. However, the transformation Eq. (19) is necessary to compensate for the phase change due to propagation. In this section, a new transformation is introduced that allows the design of a multi-output network for simultaneous decoding of all subcarriers.

From Eq. (19) one can see the propagation phase compensation is done through the exponential term. The shift of \tilde{q} in ω is used to align all the subcarriers. If all the subcarriers are going to be decoded at once, the frequency shift is not necessary. Furthermore, $2\lambda_k$ is the central frequency of subcarrier k . To decode all subcarriers at once, one needs to compensate for the phase change for all frequencies. Therefore, we rewrite transformation in the following way. Firstly, we remove the frequency shift

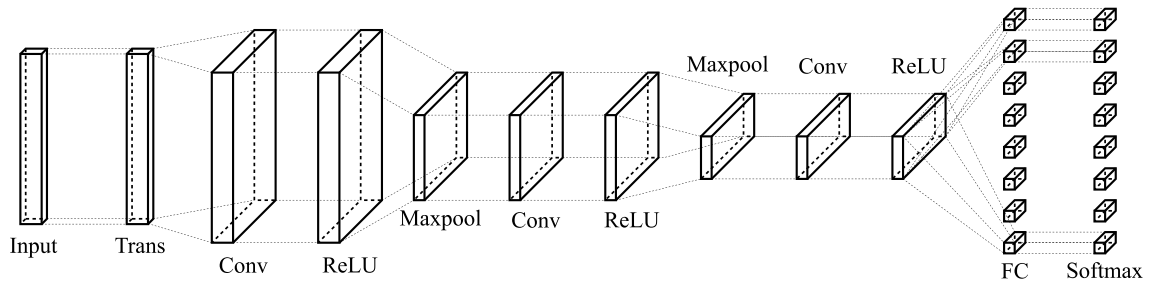


Figure 7. The conceptual schematic of the parallel network.

$$\tilde{q}_k^c(\omega, z) = \tilde{q}_k(\omega + 2\lambda_k, z) \tag{20}$$

$$= \tilde{q}(\omega + 2\lambda_k - 2\lambda_k, z) e^{j t_k(\omega + 2\lambda_k - \lambda_k)} \tag{21}$$

$$= \tilde{q}(\omega, z) e^{j t_k(\omega + \lambda_k)}. \tag{22}$$

Next, we replace λ_k with λ to compensate for the phase change for all frequencies.

$$\tilde{q}^c(\omega, z) = \tilde{q}(\omega, z) e^{j 4\lambda z(\omega + \lambda)} \tag{23}$$

$$= \tilde{q}(\omega, z) e^{-j 2\omega z(\omega - 0.5\omega)} \tag{24}$$

$$= \tilde{q}(\omega, z) e^{-j\omega^2 z}, \tag{25}$$

where $2\lambda = -\omega$ (the negative sign depends on the sign in the linear Fourier transform) and \tilde{q}^c is \tilde{q} after compensation. In principle we can apply the continuous frequency compensation to the serial scheme as well, however, the exponential calculation is computationally expensive and is not necessary for the serial case.

Now, the propagation phase is removed from the pulse, we come up with the following network scheme for multi-subcarrier simultaneous decoding. Figure 7 shows the design concept of the network. In comparison to the serial scheme, multiple fully connected layers are added in parallel to the last activation layer (ReLU) followed by softmax layers. A custom training process is used for this new scheme. The cross-entropy function is applied to all “softmax” outputs and the averaged cross-entropy is used as the loss value for calculating gradients for the next iteration. In the figure, only three convolutional layers are shown, but as the number of subcarriers increases, additional convolutional layers can be added if necessary as well as increase the number of filters in each layer.

The training of the parallel network requires more train data than the serial scheme. For the example shown here, a total of 200,000 sample pulses were generated, 90% of which was used for training and 10% for validation. The convergence of the network is similar to the serial scheme. The training and validation accuracy and loss can be found in Fig. 8. The accuracy reaches 95% within the first epoch and reaches 99% before epoch 5. The initial learning rate for the first 10 epochs is 0.002. With this learning rate, we notice big accuracy fluctuations toward later epochs. The fluctuation is reduced immediately after reducing the learning rate to 0.0002 and the accuracy reaches 99.9% within the following epoch. A small gap between training and validation is observed. Further reducing the learning rate to 0.00002 results in a further reduction in training loss but the validation loss remains the same, a slight indication of over-fitting. We speculate a larger training data can help overcome the over-fitting and further improve the accuracy and loss of the network.

The accuracy for different power ranges and individual subcarriers are shown in Fig. 9. The behaviours are rather similar to the serial network. The accuracy for different power ranges are rather close while the accuracy for the subcarriers on the edges of the spectrum is slightly lower than the majority.

The trained network of the parallel scheme occupies 128 MB of memory while the serial network occupies merely 0.5 MB. It is possible to improve the performance of the network by considering adding more hidden layers to the network or increasing the layer size. At the same time, it is also worth considering new network design concepts such as ResNet⁵⁰, InceptionNet⁵¹ and SENet⁵² to get better overall performance with smaller network sizes. As the complexity of the network increases, one may also look into more advanced training methods, such as activated gradients method⁵³, to improve the training performance for the network. In the networks presented here, all the convolutional layers contain $844N$ operations in total including $468N$ multiplications and $376N$ additions, where N is the input size. The operations in ReLU activation and max-pooling layers are negligible (about $2N$ operations). A fully connected layer has $32N$ operations with $16N$ multiplications and $16N$ additions. In the serial network, $N = 64$, but the computation is repeated M times in sequence with a total of $878 \times N \times M$ operations. In the parallel network, $N = 4096$ and only one-pass computation is required. There are M fully connected layers in the multi-output stage resulting in a total of $(846 + 32 \times M) \times N$ operations, which is about 2.8 times more operations than the serial scheme but the computation can be performed in parallel to save overall execution time.

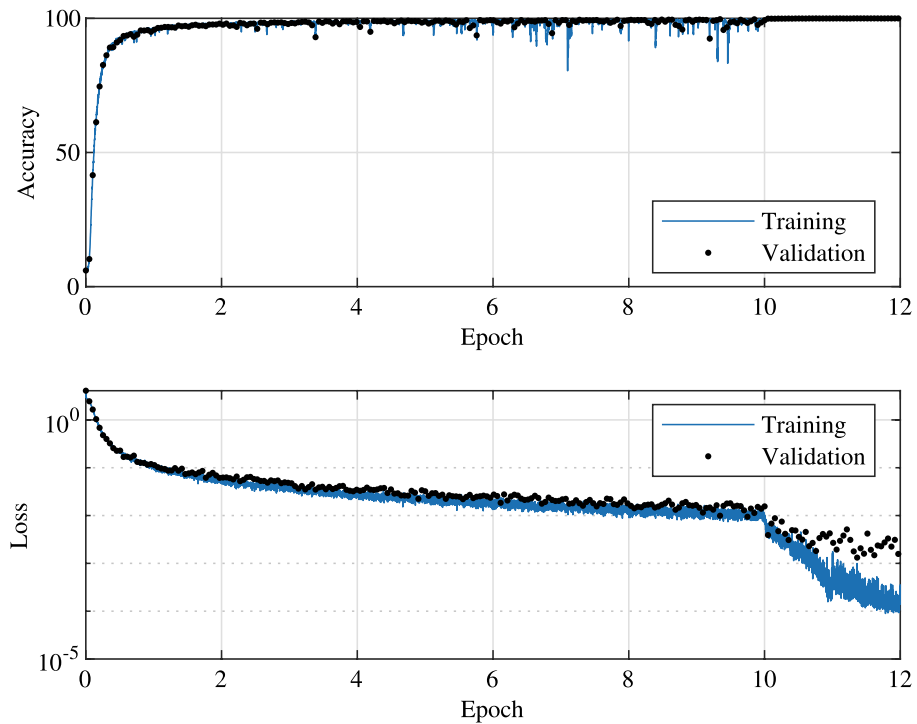


Figure 8. The accuracy and loss as functions of epochs for the parallel network scheme. An initial learning rate of 0.002 is used for epochs from 1 to 10 and increased to 0.0002 for epoch 11 and 0.00002 for epoch 12.

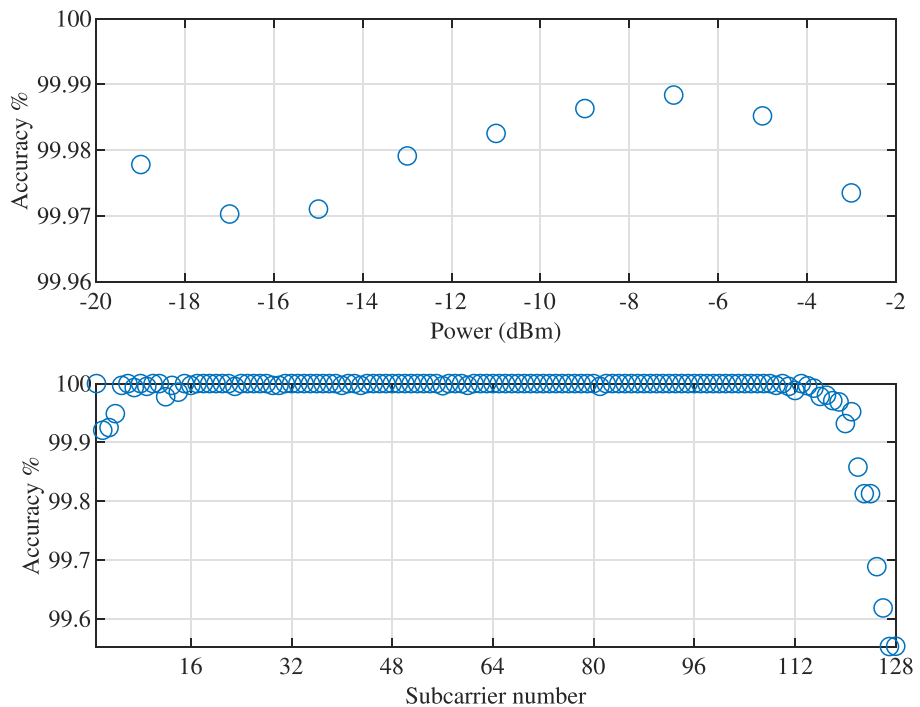


Figure 9. The accuracy for different pulse power and individual subcarrier.

Conclusion

In this paper, we proposed two conceptual CNN schemes for decoding NFDm signals. A serial scheme that sacrifices speed for small network size and a parallel scheme optimizes speed over size. The serial scheme only occupies 0.5 MB of memory space suitable for implementation on small computer chips for small users while the parallel scheme occupies 128 MB of memory which requires more expansive hardware for usages in places

such as data centres. Both network schemes have been demonstrated by training with simulated data and been able to reach more than 99.9% accuracy.

This work only demonstrates conceptual designs based on simulation of pulse propagation in optical fibres. To implement the proposed CNN scheme for practical experimentation, additional adjustments and modifications may be required. In practice, for instance, an unknown initial phase may present in the signal, which can slowly drift over time. This initial phase can be seen as a rotation of the constellation of symbol c_n . In this work, the simulated training data have zero initial phases. However, this phase information is contained inside each signal burst and thus can potentially be identified using a CNN as well. For future work, network design shall take the initial phase into account either through a separate network or adding layers to the current designs to make the network more suitable for practical use. Demonstrations of the proposed schemes using experimentally collected data will also be considered in the next stage of development.

Methods

The CNN models in this work are developed using Matlab's Deep Learning Toolbox (version R2020a), but the same results can be reproduced using other deep learning frameworks such as TensorFlow and PyTorch. In each network, the input layer is a N -row long by 2-column wide array, where each column corresponds to the real and imaginary parts of the signal. Number N is the number of sampling points across a subcarrier segment of pulse spectrum in the serial scheme or the full spectrum of the pulse in the parallel scheme. The first convolution layer has a kernel size of 3×3 , a stride step size of 1, a padding size of 1 and no dilation. The rest of the convolution layers are the same except a kernel size is reduced to 3×1 . The first max-pooling layer has a pool size of 2×2 , a stride step size of 2 and no padding. For the rest of the max-pooling layers, the pool size is reduced to 2×1 . Each fully connected layer has an output size of 16 corresponding to the 16-QAM used in this work, which is mapped to values from 0 to 15 directly and used as labels for each input. Adam optimizer was used for training the network. The decay rate of gradient and squared gradient moving averages are set to be 0.9 and 0.99, respectively. L2 regularization is applied with a factor of 0.00002. The initial learning rate is set to 0.002 and then reduced manually each time by a factor of 10 during the training process. More details can be found in corresponding sections where the network training results are discussed.

The key information of training data generation is given in the data generation section, in which, the pulse propagation over a standard single-mode optical fibre is simulated by solving the following general nonlinear Schrödinger equation using a split-step Fourier method⁵⁴,

$$\frac{\partial}{\partial z} q(z, t) + \frac{\alpha}{2} q(z, t) - \sum_{k \geq 2} \frac{i^{k+1}}{k!} \beta_k \frac{\partial^k}{\partial t^k} q(z, t) = i\gamma \left(1 + i\tau_0 \frac{\partial}{\partial t} \right) \left(q(z, t) \int_{-\infty}^{\infty} R(t') \times |q(z, t - t')|^2 dt' \right), \quad (26)$$

in which $R(t) = (1 - f_R)\delta(t) + f_R h_R(t)$, $h_R(t) = \frac{\tau_1^2 + \tau_2^2}{\tau_1 \tau_2^2} \exp(-\frac{t}{\tau_2}) \sin(\frac{t}{\tau_1})$, $f_R = 0.18$, $\tau_0 = 0.82$ fs, $\tau_1 = 12.2$ fs, $\tau_2 = 32$ fs. The fibre loss coefficient $\alpha = 0.0461$ km⁻¹, dispersion $\beta_2 = -21.7$ ps²/km, $\beta_3 = 18.6$ ps³/km and nonlinearity $\gamma = 1.1$ W⁻¹ km⁻¹. In the split-step method, 2048 sampling points are used in the time window with a sampling rate of 96 GS/s. The propagation step size is set to 10 m per step. After each span of propagation, the pulses are amplified to compensate the loss during the propagation through a scaling factor $\exp(\frac{\alpha}{2}L)$, where $L = 50$ km is the span length. A white Gaussian noise of -40 dBm is added every time the signal is amplified.

Received: 7 October 2021; Accepted: 18 April 2022

Published online: 13 May 2022

References

1. Yousefi, M. I. & Kschischang, F. R. Information transmission using the nonlinear Fourier transform, Part III: Spectrum modulation. *IEEE Trans. Inf. Theory* **60**, 4346–4369. <https://doi.org/10.1109/TIT.2014.2321155> (2014).
2. Yousefi, M. I. & Kschischang, F. R. Information transmission using the nonlinear Fourier transform, Part II: Numerical methods. *IEEE Trans. Inf. Theory* **60**, 4329–4345. <https://doi.org/10.1109/TIT.2014.2321151> (2014).
3. Yousefi, M. I. & Kschischang, F. R. Information transmission using the nonlinear Fourier transform, Part I: Mathematical tools. *IEEE Trans. Inf. Theory* **60**, 4312–4328. <https://doi.org/10.1109/TIT.2014.2321143> (2014).
4. Shannon, C. E. A mathematical theory of communication. *Bell Syst. Tech. J.* **27**, 379–423. <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x> (1948).
5. Le, S. T., Buelow, H. & Aref, V. Demonstration of 64× 0.5Gbaud nonlinear frequency division multiplexed transmission with 32QAM. In *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, 1–3 (2017).
6. Le, S. T., Aref, V. & Buelow, H. 125 Gbps pre-compensated nonlinear frequency-division multiplexed transmission. In *2017 European Conference on Optical Communication (ECOC)*, 1–3. <https://doi.org/10.1109/ECOC.2017.8346228> (2017).
7. Le, S. T. & Buelow, H. High performance NFDM transmission with b-modulation. In *Photonic Networks; 19th ITG-Symposium*, 1–6 (2018).
8. Turitsyn, S. K. *et al.* Nonlinear Fourier transform for optical data processing and transmission: Advances and perspectives. *Optica* **4**, 307. <https://doi.org/10.1364/OPTICA.4.000307> (2017).
9. Lima, I. T., DeMenezes, T. D. S., Grigoryan, V. S., O'Sullivan, M. & Menyuk, C. R. Nonlinear compensation in optical communications systems with normal dispersion fibers using the nonlinear Fourier transform. *J. Lightw. Technol.* **35**, 5056–5068. <https://doi.org/10.1109/JLT.2017.2766622> (2017).
10. Wahls, S. & Poor, H. V. Fast numerical nonlinear Fourier transforms. *IEEE Trans. Inf. Theory* **61**, 6957–6974. <https://doi.org/10.1109/TIT.2015.2485944> (2015).
11. Vaibhav, V. Numerical methods for fast nonlinear Fourier transformation, Part I: Exponential Runge–Kutta and linear multistep methods. [arXiv:1812.04701](https://arxiv.org/abs/1812.04701) [physics] (2018).

12. Chimmalgi, S., Prins, P. J. & Wahls, S. Fast nonlinear Fourier transform algorithms using higher order exponential integrators. *IEEE Access* **7**, 145161–145176. <https://doi.org/10.1109/ACCESS.2019.2945480> (2019) (Conference Name: IEEE Access.).
13. Aref, V., Le, S. T. & Buelow, H. An efficient nonlinear Fourier transform algorithm for detection of eigenvalues from continuous spectrum. In *Optical Fiber Communication Conference (OFC) 2019 (2019)*, paper M11.5, M11.5, <https://doi.org/10.1364/OFC.2019.M11.5> (Optical Society of America, 2019).
14. Vasylychenkova, A., Salnikov, D., Karaman, D., Vasylychenkov, O. G. & Prilepskiy, J. E. Fixed-point realisation of fast nonlinear Fourier transform algorithm for FPGA implementation of optical data processing. In *Nonlinear Optics and Applications XII* 11770, 111–120. <https://doi.org/10.1117/12.2588735> (SPIE, 2021).
15. Sidelnikov, O., Redyuk, A. & Sygletos, S. Equalization performance and complexity analysis of dynamic deep neural networks in long haul transmission systems. *Opt. Express* **26**, 32765–32776. <https://doi.org/10.1364/OE.26.032765> (2018) (Publisher: Optical Society of America).
16. Musumeci, F. *et al.* An overview on application of machine learning techniques in optical networks. *IEEE Commun. Surv. Tutor.* **21**, 1383–1408. <https://doi.org/10.1109/COMST.2018.2880039> (2019) (Conference Name: IEEE Communications Surveys Tutorials).
17. Schaedler, M. *et al.* Deep neural network equalization for optical short reach communication. *Appl. Sci.* **9**, 4675. <https://doi.org/10.3390/app9214675> (2019) (Number: 21 Publisher: Multidisciplinary Digital Publishing Institute).
18. Freire, P. J. *et al.* Complex-valued neural network design for mitigation of signal distortions in optical links. *J. Light. Technol.* 1–1. <https://doi.org/10.1109/JLT.2020.3042414> (2020). Conference Name: Journal of Lightwave Technology.
19. Wang, D. & Zhang, M. Artificial intelligence in optical communications: From machine learning to deep learning. *Front. Commun. Netw.* **2**, 9. <https://doi.org/10.3389/frcmn.2021.656786> (2021).
20. Pan, X. *et al.* Machine-learning-aided optical fiber communication system. *IEEE Netw.* **35**, 136–142. <https://doi.org/10.1109/MNET.011.2000676> (2021) (Conference Name: IEEE Network).
21. Xie, Y., Wang, Y., Kandeepan, S. & Wang, K. Machine learning applications for short reach optical communication. *Photonics* **9**, 30. <https://doi.org/10.3390/photonics9010030> (2022) (Number: 1 Publisher: Multidisciplinary Digital Publishing Institute).
22. Jones, R. T., Gaiarin, S., Yankov, M. P. & Zibar, D. Time-domain neural network receiver for nonlinear frequency division multiplexed systems. *IEEE Photon. Technol. Lett.* **30**, 1079–1082. <https://doi.org/10.1109/LPT.2018.2831693> (2018) (Conference Name: IEEE Photonics Technology Letters).
23. Kotlyar, O. *et al.* Unsupervised and supervised machine learning for performance improvement of NFT optical transmission. In *2018 IEEE British and Irish Conference on Optics and Photonics (BICOP)*, 1–4. <https://doi.org/10.1109/BICOP.2018.8658274> (2018).
24. Kotlyar, O., Kamalian-Kopae, M., Prilepskiy, J. E., Pankratova, M. & Turitsyn, S. K. Machine learning for performance improvement of periodic NFT-based communication system. In *45th European Conference on Optical Communication (ECOC 2019)*, 1–4. <https://doi.org/10.1049/cp.2019.1089> (2019).
25. Kotlyar, O. *et al.* Combining nonlinear Fourier transform and neural network-based processing in optical communications. *Opt. Lett.* **45**, 3462–3465. <https://doi.org/10.1364/OL.394115> (2020) (Publisher: Optical Society of America).
26. Kotlyar, O. *et al.* Convolutional long short-term memory neural network equalizer for nonlinear Fourier transform-based optical transmission systems. *Opt. Express* **29**, 11254. <https://doi.org/10.1364/OE.419314> (2021).
27. Sedov, E. V. *et al.* Neural networks for computing and denoising the continuous nonlinear Fourier spectrum in focusing nonlinear Schrödinger equation. *Sci. Rep.* **11**, 22857. <https://doi.org/10.1038/s41598-021-02252-9> (2021).
28. Sedov, E. V., Chekhovskoy, I. S. & Prilepskiy, J. E. Neural network for calculating direct and inverse nonlinear Fourier transform. *Quantum Electron.* **51**, 1118. <https://doi.org/10.1070/QEL17655> (2021) (Publisher: IOP Publishing).
29. Zhang, W. Q., Chan, T. H. & Afshar, V. S. Direct decoding of nonlinear OFDM-QAM signals using convolutional neural network. *Opt. Express* **29**, 11591. <https://doi.org/10.1364/OE.419609> (2021).
30. Capra, M. *et al.* An updated survey of efficient hardware architectures for accelerating deep convolutional neural networks. *Future Internet* **12**, 113. <https://doi.org/10.3390/fi12070113> (2020).
31. Wang, Y. E., Wei, G.-Y. & Brooks, D. Benchmarking TPU, GPU, and CPU platforms for deep learning. [arXiv:1907.10701](https://arxiv.org/abs/1907.10701) [cs, stat] (2019).
32. Zheng, W., Zhao, Y., Chen, Y., Park, J. & Shin, H. Hardware architecture exploration for deep neural networks. *Arab. J. Sci. Eng.* **46**, 9703–9712. <https://doi.org/10.1007/s13369-021-05455-4> (2021).
33. Newsroom, N. NVIDIA doubles down: Announces A100 80GB GPU, Supercharging World's Most Powerful GPU for AI Supercomputing.
34. Warren, T. Nvidia's first Ampere GPU is designed for data centers and AI, not your PC (2020).
35. Accelerator Module.
36. Shahid, A. & Mushtaq, M. A survey comparing specialized hardware and evolution in TPUs for neural networks. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*, 1–6. <https://doi.org/10.1109/INMIC50486.2020.9318136> (2020). ISSN: 2049-3630.
37. Aref, V., Le, S. T. & Buelow, H. Modulation over nonlinear Fourier spectrum: Continuous and discrete spectrum. *J. Lightw. Technol.* **36**, 1289–1295. <https://doi.org/10.1109/JLT.2018.2794475> (2018).
38. Gemechu, W. A., Song, M., Jaouen, Y., Wabnitz, S. & Yousefi, M. I. Comparison of the nonlinear frequency division multiplexing and OFDM in experiment. In *2017 European Conference on Optical Communication (ECOC)*, 1–3. <https://doi.org/10.1109/ECOC.2017.8346229> (2017).
39. Le, S. T. *et al.* Achievable information rate of nonlinear inverse synthesis based 16QAM OFDM transmission. In *ECOC 2016; 42nd European Conference on Optical Communication*, 1–3 (2016).
40. Derevyanko, S. A., Prilepskiy, J. E. & Turitsyn, S. K. Capacity estimates for optical transmission based on the nonlinear Fourier transform. *Nat. Commun.* **7**, 12710. <https://doi.org/10.1038/ncomms12710> (2016).
41. Civelli, S., Forestieri, E. & Secondini, M. Mitigating the impact of noise on nonlinear frequency division multiplexing. *Appl. Sci.* **10**, 9099. <https://doi.org/10.3390/app10249099> (2020) (Number: 24 Publisher: Multidisciplinary Digital Publishing Institute).
42. Wahls, S. Generation of time-limited signals in the nonlinear fourier domain via b-modulation. In *2017 European Conference on Optical Communication (ECOC)*, 1–3. <https://doi.org/10.1109/ECOC.2017.8346231> (2017).
43. Gui, T., Zhou, G., Lu, C., Lau, A. P. T. & Wahls, S. Nonlinear frequency division multiplexing with b-modulation: Shifting the energy barrier. *Opt. Express* **26**, 27978–27990. <https://doi.org/10.1364/OE.26.027978> (2018) (Publisher: Optical Society of America).
44. Yangzhang, X., Aref, V., Le, S. T., Bulow, H. & Bayvel, P. 400 Gbps dual-polarisation non-linear frequency-division multiplexed transmission with B-modulation. In *2018 European Conference on Optical Communication (ECOC)*, 1–3. <https://doi.org/10.1109/ECOC.2018.8535128> (IEEE, Rome, 2018).
45. Vasylychenkova, A., Pankratova, M., Prilepskiy, J., Chichkov, N. & Turitsyn, S. Signal-dependent noise for B-modulation NFT-based transmission. In *2019 Conference on Lasers and Electro-Optics Europe European Quantum Electronics Conference (CLEO/Europe-EQEC)*, 1–1. <https://doi.org/10.1109/CLEOE-EQEC.2019.8872660> (2019).
46. Wahls, S., Chimmalgi, S. & Prins, P. J. Wiener-Hopf method for b-modulation. In *Optical Fiber Communication Conference (OFC) 2019 (2019)*, paper W2A.50, W2A.50. <https://doi.org/10.1364/OFC.2019.W2A.50> (Optical Society of America, 2019).
47. Kamalian, M., Prilepskiy, J. E., Le, S. T. & Turitsyn, S. K. On the design of NFT-based communication systems with lumped amplification. *J. Lightw. Technol.* **35**, 5464–5472. <https://doi.org/10.1109/JLT.2017.2775105> (2017).
48. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) [cs] (2017).
49. Murphy, K. P. *Machine Learning: A Probabilistic Perspective* (MIT Press, 2012).
50. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. 770–778 (2016).

51. Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).
52. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. 7132–7141 (2018).
53. Liu, M., Chen, L., Du, X., Jin, L. & Shang, M. Activated gradients for deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* <https://doi.org/10.1109/TNNLS.2021.3106044> (2021).
54. Agrawal, G. P. *Nonlinear Fiber Optics* (Academic Press, 2013).

Acknowledgements

This research was supported fully by the Australian Government through the Australian Research Council (DP190102896).

Author contributions

W.Q.Z. was involved in conceptualizing the idea, the designing and simulating the CNN, analyzing the results and writing the original draft. T.H.C. and S.A.V. were involved in project management. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-022-12141-4>.

Correspondence and requests for materials should be addressed to W.Q.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022