Serial and Parallel Convolutional Neural Network Designs for NFDM Signals

Wen Qi Zhang (wenqi.zhang@unisa.edu.au)
Laser Physics and Photonic Devices Laboratories, School of Engineering, University of South Australia

Terence Chan
Institute for Telecommunications Research, University of South Australia

Shahraam Afshar V.
Laser Physics and Photonic Devices Laboratories, School of Engineering, University of South Australia

Research Article

Keywords: Serial and parallel, convolutional neural network designs, NFDM signals, decoding, hardware implementation, serial network

Posted Date: October 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-958567/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
Serial and parallel convolutional neural network designs for NFDM signals

Wen Qi Zhang\textsuperscript{1,*}, Terence H. Chan\textsuperscript{2}, and Shahraam Afshar V\textsuperscript{1}

\textsuperscript{1}Laser Physics and Photonic Devices Laboratories, School of Engineering, University of South Australia, Australia
\textsuperscript{2}Institute for Telecommunications Research, University of South Australia, Australia
*wenqi.zhang@unisa.edu.au

ABSTRACT

Two conceptual convolutional neural network designs are proposed for directly decoding NFDM signals with the consideration of hardware implementation. A serial network is designed for small network size suitable for small user applications and a parallel network is designed for speed suitable for places such as data centres. In the numerical demonstrations, the serial design only occupies 0.5 MB of memory space while the parallel design occupies 128 MB of memory but allow parallel computing. Both network designs were trained with simulated data and able to reach more than 99.9% accuracy.

Introduction

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\textsuperscript{1–4}. It has already been demonstrated experimentally that NFT can outperform conventional linear Fourier transform-based systems\textsuperscript{5–7}. However, NFT-based communication methods have not yet reached the maturity level of conventional methods. One aspect is the lack of hardware implementation of NFT. Only very recently, hardware implementation was conceptually proposed on an FPGA platform\textsuperscript{8}. However, due to the complexity of NFT algorithms, implementing and optimizing the hardware can take years of effort before any products can be built for practical use.

On the other end, it has been shown that a convolutional neural network (CNN) can be used to directly decode nonlinear frequency division multiplexing (NFDM) signals without actually performing NFT\textsuperscript{9}. Due to the popularity of machine learning, hardware implementation of deep neural networks itself has become an active research area with supports from mega-corporations such as Google and Nvidia\textsuperscript{10–16}. It is an opportunity to take the advantage of the rapid development of deep learning hardware and use it for NFT applications.

In the previous work\textsuperscript{9}, the network design was not used efficiently, especially for signals with a large number of subcarriers, since for every subcarrier, we need a separate network. Furthermore, the previous work only demonstrated networks for a fixed signal power without considering fibre loss and in the case of lump amplification. In this paper, we are going to show two network design ideas with hardware implementation in mind. A serial design aimed at small network size and a parallel design aimed at performance. We will also demonstrate the networks with variable signal power and in case of loss and lump amplification.

The paper is divided into 5 sections apart from the Introduction. In section 2, a quick introduction to nonlinear Fourier transform and nonlinear frequency division multiplexing (NFDM) signals is given. Section 3 gives the basic parameters and description of the generation training and validation data. Section 4 and 5 discuss the serial and parallel network design respectively followed by a conclusion in Section 6.

Modulation of continuous nonlinear spectrum

Transforming a temporal signal $q(t)$ in the nonlinear Fourier domain equals solving the following differential equation\textsuperscript{1}

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

with the initial condition:

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

(1)
where \( v_r \) is the derivative of \( v \) and \( \lambda \) is the nonlinear frequency. Two time invariant coefficients \( a(\lambda) \) and \( b(\lambda) \) can be found using the following limits:\(^3\)

\[
a(\lambda) = \lim_{t \to \infty} v_1^2(t, \lambda) e^{i\lambda t}, \quad b(\lambda) = \lim_{t \to \infty} v_2^2(t, \lambda) e^{-j\lambda t}.
\]

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

\[
Q(\lambda) = \begin{cases} 
\frac{b(\lambda)}{a(\lambda)} & \text{for } \lambda \in \mathbb{R}, \\
\frac{b(\lambda)}{a'(\lambda)} & \text{for } \lambda \in \mathbb{C}^+ \text{ and } a(\lambda) = 0,
\end{cases}
\]

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 with a replacement of the linear spectrum with a nonlinear spectrum\(^3\) and becoming popular in recent years\(^6\)–\(^21\). 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\(^7\)–\(^26\).

The modulation can often be expressed as

\[
s(\lambda) = \sum_n c_n w(\lambda),
\]

where \( n \) is the number of subcarriers, \( c_n \) is the complex data symbols, \( w_n(\lambda) \) is the carrier wave. Conventionally, \( \text{sinc} \) function is used as \( w(\lambda) \).\(^7\) Other choices of functions such as raised cosine or flat-top window with cardinal sine carrier waveform have also been proposed\(^23\). 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 \( q(t) \).\(^17\) 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 taken\(^22\).

Inspired by the work\(^23\), we used the following carrier wave functions for this work.

\[
w_n(t) = \text{sinc} \left( \frac{4t}{T_0} \right) \cdot \exp \left[ -2 \left( \frac{t}{T_0} \right)^2 - i n \pi \cdot \frac{8t}{T_0} \beta \right]
\]

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

where \( T_0 \) denotes the width to the 4th zero of the sinc function [note: here \( \text{sinc}(t) = \sin(\pi t)/ (\pi t) \)]. \( \tau \) 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, \( \beta \) 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 \( \pm 2\lambda \) (or \(-2\lambda \) depending on the choice of the sign in the Fourier transform), where \( w_n(\omega) \) is the linear Fourier transform on \( w_n(t) \).

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

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 propagation loss of 0.2 dB/km, group velocity...
displacement of -21.7 ps²/km (β₃ = 18.6 ps³/km) and nonlinearity of 1.1 W⁻¹km⁻¹. A maximum of 10 optical fibre spans with 50 km per span is 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 to 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 amplification. A quadrature amplitude modulation is used (16-QAM) for encoding subcarriers with direct mapping of numbers 0 to 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 dBm 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 develop and used the approximated energy calibration model shown in 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

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

(9)

where \( A \) is a real scalar 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 dBm 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

\[
E = \frac{1}{\pi} \sum_i \ln \left( 1 + A^2 |Q_i|^2 \right) A\lambda
\]  

(10)

Rearranged the equation, we have

\[
E \frac{\pi}{A\lambda} = \sum_i \ln \left( 1 + A^2 |Q_i|^2 \right)
\]  

(11)

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

(12)

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

\[
\ln E + \ln \frac{\pi}{A\lambda} = \ln \ln C
\]  

(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 \( \bar{Q} \) such that

\[
E \frac{\pi}{A\lambda} = \ln \left( \prod_i \left( 1 + A^2 |\bar{Q}|^2 \right) \right)
\]  

(14)

\[
= \ln \left( 1 + A^2 |\bar{Q}|^2 \right)^n
\]  

(15)

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

(16)

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

\[
A' |1| = A |\bar{Q}| = \sqrt{\frac{\pi \cdot \ln C}{\Lambda}} - 1.
\]  

(17)

And we choose a few points \( E' \) within the energy range \([E_{\text{min}}, E_{\text{max}}]\) 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 \( E_{\text{avg}} \) using randomly generated \( Q(\lambda) \). Depending on the
choice of carrier wave function, QAM format, number of subcarriers, etc., $E_{\text{avg}}$ shall differ from $E'$. Figure 1 shows an example. Finally, a least square fit is applied to $\ln E_{\text{avg}}$ and $\ln E'$,

$$ \begin{pmatrix} p_1 \\ p_0 \end{pmatrix} = \text{pinv} \left( \begin{bmatrix} \ln E_{\text{avg}} & J \end{bmatrix} \right) \times \ln E', $$

(18)

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

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

The serial CNN design

With the original work\textsuperscript{9}, to decode each subcarrier, a transformation need to be performed on the received signal to remove the propagation induced temporal shift and phase change. Then, each subcarrier is decoded using its own corresponding network. This process is rather 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 nonlinear 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 nonlinear frequency domain, hence greatly reduce the parameter space of the network.

Figure 2 illustrate 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.

With these ideas in mind, we design 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).

$$ \tilde{q}_k(\omega, z) = \tilde{q}(\omega - 2\lambda_k, z)e^{j\tau_k(\omega - \lambda_k)}, $$

(19)

where $\lambda_k$ is the central nonlinear frequency of subcarrier $k$, $\tau_k = 4\lambda_k z$. This equation shifts the central frequency of subcarrier $k$ to subcarrier 0 and removes the phase change due to propagation from the nonlinear spectrum\textsuperscript{9}. 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 to 15.
Figure 2. An example signal is shown in the linear frequency domain compared with subcarriers waves in the nonlinear frequency domain. 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.

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.

The design 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 signal. Eq. 19 is implemented into the
network as a transform layer. All the convolutional layers have 64 kernels with a size of $3 \times 3$ for the first layers and $3 \times 1$ for the rest of the layers.

![Diagram of the serial network](image)

**Figure 4.** The conceptual design 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.

The network is trained using the Adam optimizer with L2 regularization. 10,000 simulated pulses were generated within which 99% is used for training and 1% is used for validation. 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 function. 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 learning rate for the first 10 epochs is set to 0.002 and then reduced to 0.0002 for the next 10 epochs. The accuracy approaches 99.9% after reducing the learning rate. 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.

![Graphs of accuracy and loss](image)

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

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 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.

![Graph showing the accuracy for different pulse power and individual subcarrier.](image)

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

This network design 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 design**

The serial design 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 purpose a parallel network design where all subcarriers can be decoded at once.

In serial design, 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 $\omega$ 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

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

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

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

Next, we replace $\lambda_k$ with $\lambda$ to compensate for the phase change for all frequencies.

$$q^*(\omega, z) = q^*(\omega, z)e^{j\lambda_k(\omega + \lambda)}$$

$$= q^*(\omega, z)e^{-j\lambda_0(\omega - 0.5\omega)}$$

$$= q^*(\omega, z)e^{-j\omega^2z}.$$
where $2\lambda = -\omega$ (the negative sign depends on the sign in the linear Fourier transform) and $\tilde{q}$ is $q$ after compensation. In principle we can apply the continuous frequency compensation to the serial design 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 design for multi-subcarrier simultaneous decoding. Figure 7 shows the design concept of the network. In comparison to the serial design, 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 design. 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.

![Figure 7. The conceptual design of the parallel network.](image)

The training of the parallel network requires more train data than the serial design. 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 design. 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 accuracies for different power ranges are very 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 design occupies 128 MB of memory while the serial design 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 increase the layer size. At the same time, it is also worth considering new network design concepts such as ResNet, InceptionNet and SENet to get better overall performance with smaller network size. In 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 design, $N=64$, but the computation is repeated $M$ times in sequence with a total of $878 \times N \times M$ operations. In the parallel design, $N=4096$ 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 design but the computation can be performed in parallel to save overall execution time.

**Conclusion**

In this paper, we proposed two conceptual CNN designs for decoding NFDM signals. A serial design that scarifies speed for small network size and a parallel design optimizes speed over size. The serial design only occupies 0.5 MB of memory space suitable for implementation on small computer chips for small users while the parallel design occupies 128 MB of memory which requires more expensive hardware for usages in places such as data centres. Both network designs were demonstrated by training with simulated data and were able to reach more than 99.9% accuracy.

The simulated training data have zero initial phases. However, in the experiments, an unknown initial phase may present and slowly drift over time. This initial phase can be seen as a rotation of the constellation of symbol $c_n$. This information is contained inside each signal burst and thus can be identified. For future work, network design shall take the initial phase
Figure 8. The accuracy and loss as functions of epochs for the parallel network design. 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.

into account either through a separate network or adding layers to the current designs to make the network more suitable for practical use.

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 first convolution has a kernel size of 3 times 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 times 1. The first max-pooling layer has a pool size of 2 times 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. 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 detail can be found in corresponding sections where the network training results are discussed.

The key information of training data generation is given in Section 3, 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 method\(^\text{33}\),

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

in which \(R(t) = (1-f_R)\delta(t) + f_R h_R(t), h_R(t) = \frac{\tau_1^2 + \tau_2^2}{\tau_1 \tau_2} \exp(-\frac{t}{\tau_1}) \sin\left(\frac{t}{\tau_1}\right), f_R = 0.18, \tau_0 = 0.82 \text{ fs}, \tau_1 = 12.2 \text{ fs}, \tau_2 = 32 \text{ fs}\). The fibre loss coefficient \(\alpha\), dispersion \(\beta\)’s and nonlinearity \(\gamma\) are given in Section 3. 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 meters per step.

References

1. Yousefi, M. I. & Kschischang, F. R. Information Transmission Using the Nonlinear Fourier Transform, Part III: Spectrum Modulation. *IEEE Transactions on Inf. Theory* 60, 4346–4369, DOI: 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 Transactions on Inf. Theory* 60, 4329–4345, DOI: 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 Transactions on Inf. Theory* 60, 4312–4328, DOI: 10.1109/TIT.2014.2321143 (2014).

4. Shannon, C. E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* 27, 379–423, DOI: 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, DOI: 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. Vasilychenkova, A., Salnikov, D., Karaman, D., Vasilychenkov, 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*, vol. 11770, 111–120, DOI: 10.1117/12.22588735 (SPIE, 2021).

9. Zhang, W. Q., Chan, T. H. & Afshar V., S. Direct decoding of nonlinear OFDM-QAM signals using convolutional neural network. *Opt. Express* 29, 11591, DOI: 10.1364/OE.419609 (2021).

10. Capra, M. et al. An Updated Survey of Efficient Hardware Architectures for Accelerating Deep Convolutional Neural Networks. *Future. Internet* 12, 113, DOI: 10.3390/fi12070113 (2020).

11. Wang, Y. E., Wei, G.-Y. & Brooks, D. Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. *arXiv:1907.10701 [cs, stat]* (2019). ArXiv: 1907.10701.

12. Zheng, W., Zhao, Y., Chen, Y., Park, J. & Shin, H. Hardware Architecture Exploration for Deep Neural Networks. *Arab. J. for Sci. Eng.* 46, 9703–9712, DOI: 10.1007/s13369-021-05455-4 (2021).

13. Newsroom, N. NVIDIA Doubles Down: Announces A100 80GB GPU, Supercharging World’s Most Powerful GPU for AI Supercomputing.

14. Warren, T. Nvidia’s first Ampere GPU is designed for data centers and AI, not your PC (2020).

15. Accelerator Module.

16. 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, DOI: 10.1109/INMIC50486.2020.9318136 (2020). ISSN: 2049-3630.
17. Aref, V., Le, S. T. & Buelow, H. Modulation Over Nonlinear Fourier Spectrum: Continuous and Discrete Spectrum. *J. Light. Technol.* **36**, 1289–1295, DOI: 10.1109/JLT.2018.2794475 (2018).

18. 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, DOI: 10.1109/ECOC.2017.8346229 (2017).

19. 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).

20. Derevyanko, S. A., Prilepsky, J. E. & Turitsyn, S. K. Capacity estimates for optical transmission based on the nonlinear Fourier transform. *Nat. Commun.* **7**, 12710, DOI: 10.1038/ncomms12710 (2016).

21. Civelli, S., Forestieri, E. & Secondini, M. Mitigating the Impact of Noise on Nonlinear Frequency Division Multiplexing. *Appl. Sci.* **10**, 9099, DOI: 10.3390/app10249099 (2020). Number: 24 Publisher: Multidisciplinary Digital Publishing Institute.

22. 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, DOI: 10.1109/ECOC.2017.8346231 (2017).

23. 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, DOI: 10.1364/OE.26.027978 (2018). Publisher: Optical Society of America.

24. 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, DOI: 10.1109/ECOC.2018.8535128 (IEEE, Rome, 2018).

25. Vasylchenkova, A., Pankratova, M., Prilepsky, 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, DOI: 10.1109/CLEOE-EQEC.2019.8872660 (2019).

26. 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, DOI: 10.1364/OFC.2019.W2A.50 (Optical Society of America, 2019).

27. Kamalian, M., Prilepsky, J. E., Le, S. T. & Turitsyn, S. K. On the Design of NFT-Based Communication Systems With Lumped Amplification. *J. Light. Technol.* **35**, 5464–5472, DOI: 10.1109/JLT.2017.2775105 (2017).

28. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* (2017). ArXiv: 1412.6980.

29. Murphy, K. P. *Machine Learning: A Probabilistic Perspective* (MIT Press, 2012).

30. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. 770–778 (2016).

31. 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).

32. Hu, J., Shen, L. & Sun, G. Squeeze-and-Excitation Networks. 7132–7141 (2018).

33. 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 statement**

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.

**Additional information**

**Competing Interests:** The authors declare no competing interests.