scientific reports

Check for updates

OPEN

A photonic complex perceptron for ultrafast data processing

Mattia Mancinelli✉**, Davide Bazzanella, Paolo Bettotti & Lorenzo Pavesi**

In photonic neural network a key building block is the perceptron. Here, we describe and demonstrate a complex-valued photonic perceptron that combines time and space multiplexing in a fully passive silicon photonics integrated circuit to process data in the optical domain. A time dependent input bit sequence is broadcasted into a few delay lines and detected by a photodiode. After detection, the phases are trained by a particle swarm algorithm to solve the given task. Since only the phases of the propagating optical modes are trained, signal attenuation in the perceptron due to amplitude modulation is avoided. The perceptron performs binary pattern recognition and few bit delayed XOR operations up to 16 Gbps (limited by the used electronics) with Bit Error Rates as low as 10^{-6}.

Photonic Neural Networks (PNN) are radically changing the benchmark of complexity and speed of computation[1,2]. Photonic Integrated Circuits (PICs) can be used in *beyond von-Neumann* architecture to perform logical operations more complex than the boolean primitives (e.g., speech[3] and image recognition[1,4], signal recovery[5,6], object classification[7]). For such complex operations, the advantages of photonics (multiwavelength, speed, low power) overcomes its limitations (mainly system complexity, power losses and footprint). An in depth description of the basics of a PNN can be found in recent review papers[8–10]. Generally, PNNs can be classified into two main categories: feed forward (FFN) and reservoir (RCN) networks. The former, mostly used in computer science as the basis of deep learning algorithms, allows the optimization of the network (in terms of weights and biases) through a deterministic algorithm (e.g., the gradient descent or the back-propagation). Despite their optimized structure, FFNs are not designed to work with time dependent signals and are hardly implemented in high speed signal processing (e.g., to correct high bandwidth optical signals distorted by nonlinear propagation effects or to analyze correlation between signals distant in time). Several papers demonstrated optical implementation of FFN in PIC as optical accelerator[7,11], since matrix multiplication is a fundamental operation in FFNs which fully exploits the benefits of optics: high speed, low power consumption and inherent parallelism. More recently, innovative approaches that make use of frequency combs to realize deep and convolutional PNN have been reported[2,12].

For the analysis of time dependent signals, recurrence is a nearly mandatory property of the network, as it enriches the network description and unveils (nonlinear) relations between retarded signals (bits) by the network memory. Recurrence greatly increases the complexity of the network dynamics and prevents the possibility of a detailed description of the instantaneous network state. Yet, it is heuristically demonstrated that if the reservoir computer (RC) is forced to work "at the edge of chaos", then the network is able to effectively compute complex tasks[13]. On this line, RC is one of the network paradigms investigated since the first experimental implementation of PNN[14]. Several papers demonstrated photonics RC made of either a single[15] or multiple nodes[16] or using passive as well as active PNN[3,17]. RC highly relaxes the complexity of the training phase as the readout is a simple linear projection from a (sub)set of the RC node states. However, the randomness of its internal states can limit the RC performance when the number of nodes is limited by the system scalability. In fact, in integrated optics implementation, recurrence in spatially distinct RC nodes suffers from propagation losses. Under these conditions, a PNN can benefits from constructing internal states by mixing information from adjacent bits and from training these states by using the additional degree of freedom provided by the phase of the optical wave in the complex domain.

For both kinds of PNNs, a key element is the multiply-accumulate (MAC) unit with a perceptron scheme[18]. During the training, this component weights the input signals to draw a linear decision boundary and to achieve the desired task. Here, we present a complex-valued photonic perceptron that combines time and space multiplexing to act as a non-linear classifier on data encoded in the optical domain. We maintain the complex nature of the propagating optical mode even during the training stage. We demonstrate a fully passive optical perceptron where only the phases of the modes are learned which avoids an a-priori optical power loss due to amplitude modulation. The perceptron makes optical computation at ultrafast speed and solves several logic tasks. Basically,

NanoLab, Department of Physics, University of Trento, Via Sommarive 14, 38123 Trento, Italy. ✉email: mattia.mancinelli@unitn.it
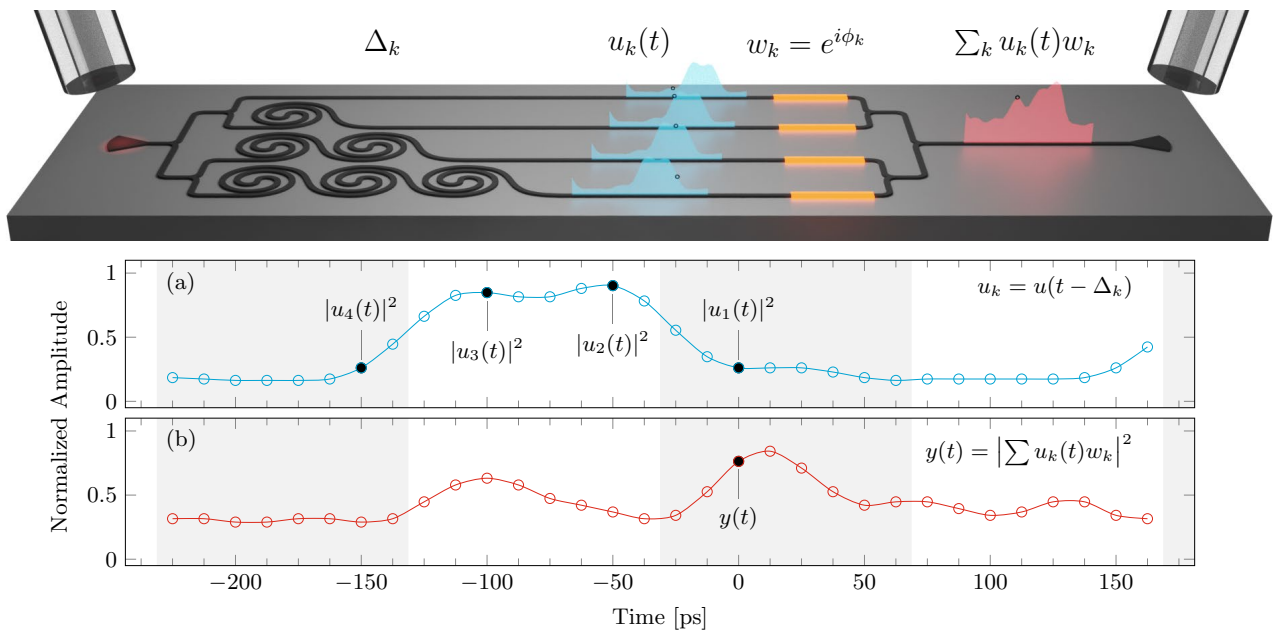
Figure 1. (Top) Sketch of the integrated photonic circuit which performs as a complex perceptron. Gratings are used to couple the light in and out of the chip. The red spot on the left shows the input signal. A 1×4 splitter distributes the input signal to four delay lines, realized by spirals (each spiral adds a delay of Δ_t). Then, thermal phase modulators (yellow segments) allow controlling the relative phases ($w_k = e^{i\phi_k}$) of the signals ($u_k(t)$, blue lineshapes). These are then summed by combiners and the resulting signal ($y(t)$, red lineshape) is output via a grating. (Bottom) (**a**) Input signal versus time. The black circles show the sampling of the input signal done by the four delay lines. (**b**) Perceptron output versus time. This signal is the one that results after the photodetector. The dot shows the time at which the complex sum is performed.

the perceptron processes time dependent input bit sequences by broadcasting them into a small number of delay lines (waveguides, WG). The perceptron is trained by optimizing the values of the relative phase of the signal in the various WGs while their amplitudes are fixed by the delay lines losses. The perceptron performs pattern recognition and delayed XOR task up to 16 Gbps (limited by the testing electronics) with Bit Error Rates (BERs) as low as 10^{-6} (that is the statistical limit of the sequences provided to the PNN). Our perceptron is integrated in silicon photonics and compatible with CMOS technology, and all its parameters are trained by using a particle swarm algorithm[19].

The complex perceptron

The integrated version of the complex perceptron is schematically shown in Fig. 1. It is composed by an input grating coupler for TE (Transverse Electric, in plane) polarization connected to a 1×4 beam splitter composed of cascaded 1×2 multimode interferometers (MMIs). Therefore, the input signal $u(t)$ is split in four copies ($u_k(t)$, k = 1,···,4) that propagate in the four WGs. The waveguides are spiraled to realize delay lines and, thus, to retard copies of the input signal by an integer multiple of $\Delta_t = 50$ ps. Thus, in each WG (k = 1,···,4) a delayed signal $u_k(t) = u(t - (k - 1)\Delta_t)$ is propagating. Thermal heaters (yellow lines) are placed above the WGs to heat them and, therefore, impart a given phase ϕ_k to the delayed input signals (phase shifters). In this way, phase encoded weights, $w_k = \exp(i\phi_k)$, are attributed to each delayed copy. Finally, the weighted delayed copies $w_k u_k$ are coherently summed by a 4×1 combiner realized by cascaded 2×1 MMI. At the output, the signal $\sum_{k=1}^{4} w_k u_k(t)$ is collected by a fiber and detected by a photodetector. At the detector, the non-linear transformation ($|\cdot|^2$) is performed on the output signal. The time dependent electrical signal $y(t) = |\sum_{k=1}^{4} w_k u_k(t)|^2$ is the output of the perceptron, i.e., the perceptron prediction. In a nut-shell, the effect of the delay lines is to populate the perceptron input layer by mixing information coming from adjacent bits and the complex perceptron performs logical operations by modulating the interference between different bits, via the phase controls.

To see the action of the phase control, let us consider three signals u_1, u_2, u_3 after the delay stages, which have been sampled from a binary sequence. Two signals are sampled from the same input bit (e.g., $u_2 = u_3$). This might happen whenever $(\Delta_t)^{-1}$ is faster than the input bit-rate. If we simplify the discussion by assuming a zero phase delay on the first delay line, then the perceptron output, y, is

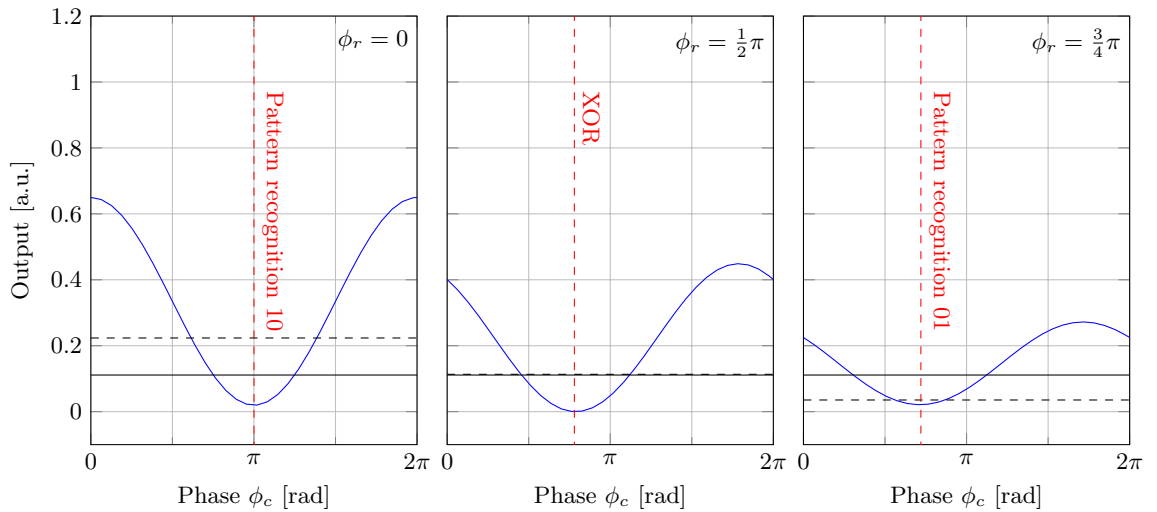$$y = \left| u_1 w_1 + u_2 w_2 + u_3 w_3 \right|^2, \tag{1}$$

Figure 2. The perceptron prediction for several input states as a function of the common phase ϕ_c between inputs u_1 and u_2, u_3 for three values of the relative phase ϕ_r between u_2 and u_3. Red dashed lines indicate the optimal common phase required to solve the particular task indicated on the lines. The values of $\{u_1, u_2, u_3\}$ are respectively: 1, 1, 1 (blue line), 1, 0, 0 (continuous horizontal line), and 0, 1, 1 (dashed horizontal line).

where $w_1 = a_1 e^{i\phi_1} = 1$; $w_2 = a_2 e^{i\phi_2}$ and $w_3 = a_3 e^{i\phi_3}$. The parameter a_k accounts for the different delay line losses. If we introduce, $\phi_c = \phi_2$ and $\phi_r = \phi_3 - \phi_2$ as the common and the relative phases of the other two signals and $\gamma = a_3/a_2$. Then,

$$y = \left| u_1 + a_2 u_2 e^{i\phi_c}(1 + \gamma e^{i\phi_r}) \right|^2. \tag{2}$$

Assuming a constant ϕ_r, we can simplify further by introducing a complex valued constant $\eta = a_2(1 + \gamma e^{i\phi_r})$:

$$y = \left| u_1 + \eta u_2 e^{i\phi_c} \right|^2, \tag{3}$$

that is, the prediction is given by the interference between u_1 and u_2. Here, it is also highlighted the role of ϕ_r, which controls the complex amplitude of the interfering signal. Therefore, despite only the phases of the weights applied to u_k are controlled, the interplay between the different phase delays among the various signals induces a rich combination of the delayed signals in the complex perceptron output.

As an example of use of the complex perceptron, we simulate the simple model of Eq. (3) and apply it to two binary tasks: two bits pattern recognition and XOR task. Results are shown in Fig. 2, where the various panels report the output signal y as a function of ϕ_c for three different ϕ_r values. The three possible bit combinations, e.g., (10), (01), and (11), are represented by the black dotted, black continuous, and blue continuous line, respectively.

The trivial case of both bits zero produces identically null output and it is not shown. The blue line shows that y results from the sinusoidal interference among the bits when they both are different from zero. By choosing the proper values for ϕ_c and ϕ_r, the complex perceptron is able to solve both the XOR and the non-trivial pattern recognition task. The red, vertical dotted lines identify the phase values where the prediction solves the various tasks. Let us note the important role of ϕ_r: Fig. 2 highlights that ϕ_r modulates the amplitude of the interference and the level of the black dashed line. When $\phi_r = 0$, the system is only able to solve the XOR task with unbalanced high levels and to find the pattern 10.

Results

The complex perceptron device creates $N = 4$ delayed input copies $u_k(t) = \{u(t), \ldots, u(t - (k-1) \cdot \Delta_t), \ldots, u(t - (N-1) \cdot \Delta_t)\}$ that populates the feature time series x_k. The perceptron applies the complex weights $w_k = \{a_1 e^{i\phi_1}, \ldots, a_N e^{i\phi_N}\}$ and performs the sum and non-linear transformation

$$y(t) = \left| \sum_{k=1}^{N} u_k(t) w_k \right|^2, \tag{4}$$

to produce the predictor $y(t)$. In the actual implementation, amplitudes are fixed by the propagation losses in the spirals and only the phases are trained. The nominal differential delay and amplitudes are $\Delta_t = 50$ ps and $a_k^2 = \{1, 0.58, 0.34, 0.2\}$. Dispersive effects in the spirals are negligible since the dispersion length $L_D = T_0^2/|\beta_2|$ for a 20 Gbps signal is estimated more than 780 m. A memory of 3 bits, for a binary input signal, is reached at
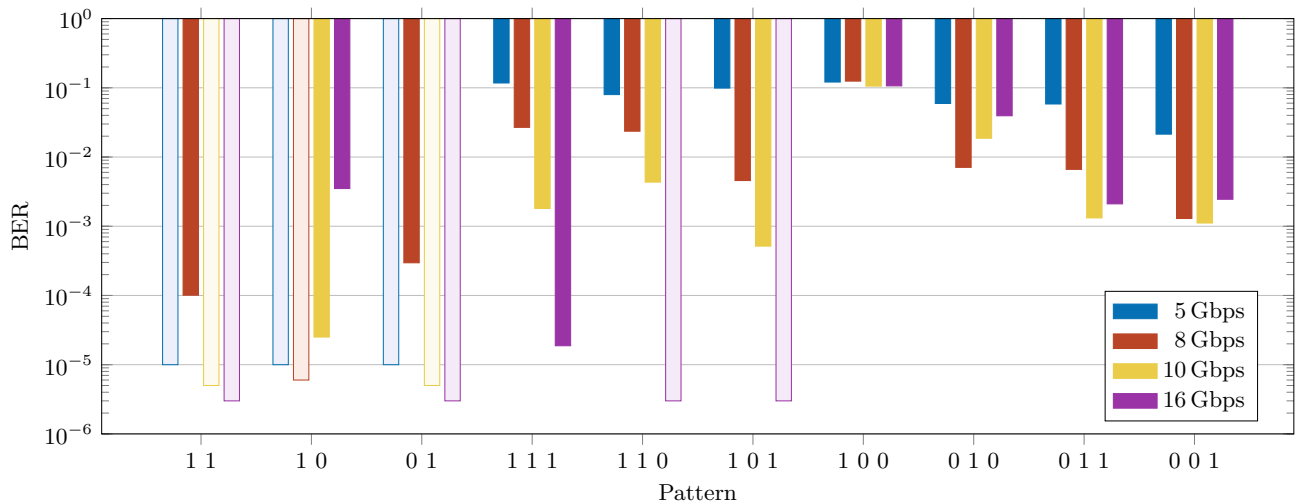
Figure 3. Measured BER as a function of the target symbol for the pattern recognition task. The different colored bars refer to the different bit-rates reported in the inset. Transparent bars refer to a BER value equal to the statistical limit (i.e., error free calculation). This value is 10^{-5}, 6×10^{-6}, 5×10^{-6}, 3×10^{-6} for a bit-rate of 5, 8, 10, 16 Gbps, respectively.

the design bit-rate of 20 Gbps. The output at any given time is the coherent weighted sum of the four delayed copies of the input, non-linearly transformed by the detector.

Pattern recognition task. The first task we consider is the recognition of a pattern of 2 or 3 bits in a pseudo-random bit sequence (PRBS) with a Non Return to Zero (NRZ) modulation format. This recognition task requires memory. The perceptron has to output a high state when the target pattern is detected and a low state for all the other cases. Causality imposes the perceptron to "*wait-and-see*" all the pattern bits before outputting the predictor. For instance, in the case of a 3-bit pattern, y is delayed by 2 bits and the prediction is aligned with the last bit in the pattern. The NRZ modulation prevents the device to recognize the symbols with all null bits since the conservation of energy requires the predictor to be identically null, too.

A Particle Swarm (PSW) algorithm[19] has been used to train the perceptron (see "Methods" section). The training has been performed acquiring, for each iteration, a 2 μs long sequence that corresponds to 3.2×10^4 bits at 16 Gbps, while testing has been done on 10 sequences of the same length. During both these phases, the algorithm updates the best sampling and threshold values, as it is typical in telecom systems. In this way, slow drifts due to environmental noise have a smaller impact on the performance of the device during the testing phase.

After training, the perceptron performances expressed as the bit error rate (BER) figure at several input signal bit-rates are reported in Fig. 3. Excellent performance is reached for the 2-bit pattern at all bit-rates. For the case of the 3-bit patterns, the best performance is achieved at either 10 or 16 Gbps, that are the bit-rates closer to the design rate of 20 Gbps (as defined by Δ_t). The most demanding pattern in terms of memory is "100" because the perceptron has to store the energy of the 1 and release it after two bit slots to output a high level. We call the rightmost bit of the sequence the reference bit, as it is the bit corresponding to the time the device has to output its predictor.

Figure 4 provides a physical insight on the perceptron operation. The assigned task is to recognize the "10" pattern at 16 Gbps. Figure 4a reports the distributions of the input levels. The level of the reference bit is partially affected by the value of the previous one, so that the distributions of "00" and "10" symbols have different mean. In these cases, the zero-level signal slightly depends on the level of the bit in the past. This phenomenon is called *intersymbolic interference* and arises from the finite bandwidth of the setup. On the other hand, the distributions of 1s and 0s of the reference bits are well separated, meaning that the signal information is well conserved. This is confirmed by the time trace in Fig. 4b, where the blue continuous line shows the clearly separated high and low levels of the input signal. In this figure, the signal at the output of the perceptron (predictor) is shown as the red dotted line. Here, the red dots indicate the best sampling time. As can be noted, the perceptron is able to solve the task for all the instances. The action of the trained perceptron on the input signal is visible in Fig. 4c: the distribution of the levels are shifted so that the classes "00", "01", and "11" are well separated from the class "10". The last is the only one above the threshold (vertical dashed line). This task highlights the importance of the perceptron memory because it successfully produces a high output even in the presence of a low input state of the reference bit.

Delayed XOR task. We use the n-bit delayed XOR task to investigate the node memory and its non-linear transformation capability[20]. The perceptron has to output the result of the XOR operation between a bit and the n-th previous bit.

Figure 5a shows that, with 1-bit delayed XOR operation, the perceptron has enough memory and non-linearity to perform error-free operation up to 8 Gbps and it achieves a BER $\sim 10^{-3}$ at 16 Gbps. An example of
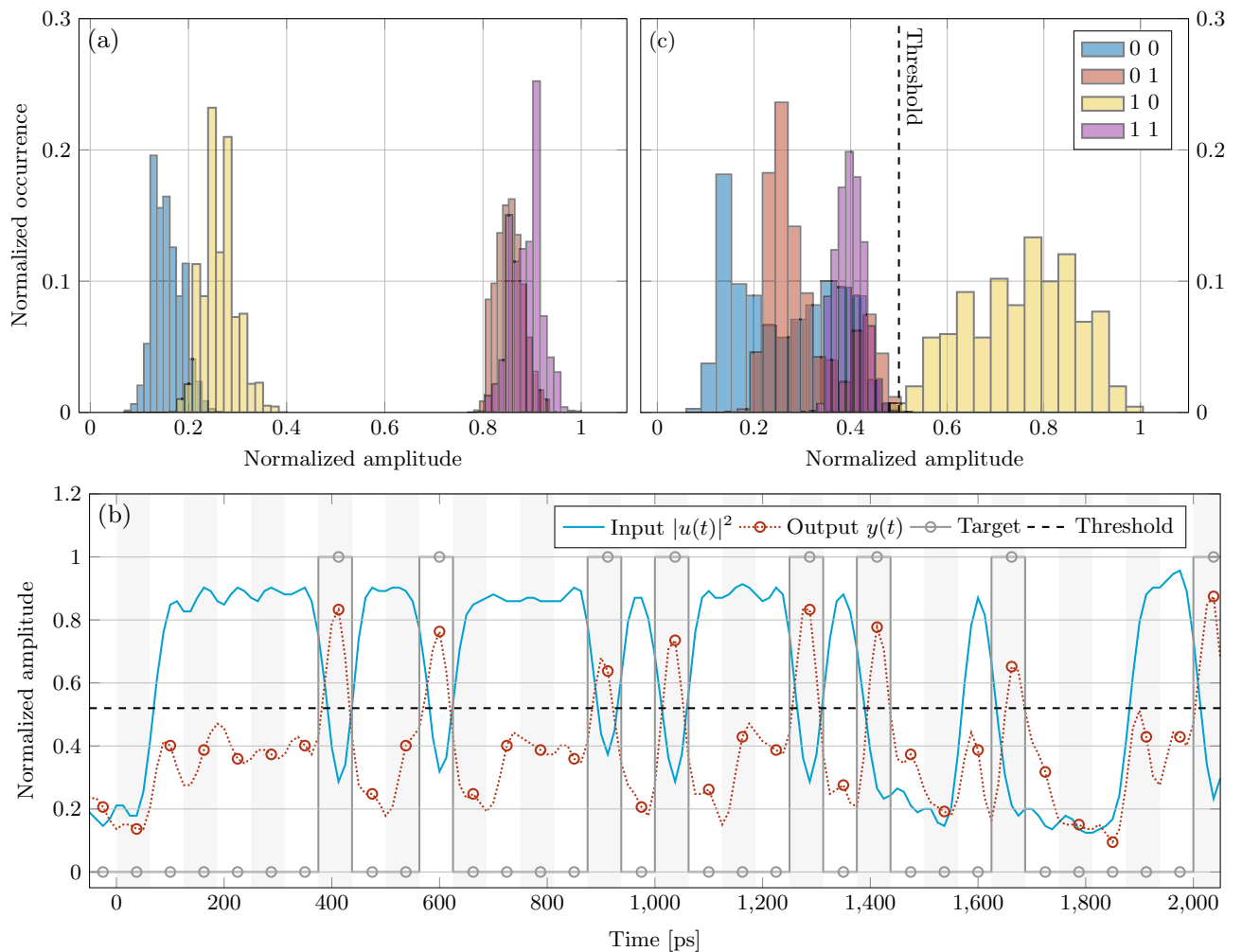
Figure 4. Perceptron trained to recognize the "10" pattern at 16 Gbps. (**a**) Input optical signal levels distribution of the 2nd bit of the 2-bit patterns given in the inset. (**b**) Time traces for the input signal $u(t)$ (blue line) and for the perceptron output $y(t)$ (red dashed line). Dashed black line is the optimal threshold. Gray circles refer to the target response, red circles indicate the best samples at the perceptron output. Vertical stripes indicate the bit slots. (**c**) Output signal levels distribution of the 2nd bit of the 2-bit patterns given in the inset. The vertical dashed line marks the optimal threshold.

the output of the perceptron at 16 Gbps is shown in Fig. 5b where the red dots show the best sampling to achieve the desired task with the optimal threshold level indicated by a black dashed horizontal line. The nonlinear transformation of the input is visible by looking at the output level histograms (Fig. 5c), which show the way the perceptron separates the output levels to perform the desired task (bit-rate 16 Gbps).

An important action during the perceptron training phase is the selection of the best sampling time t_S, i.e., the best time within the bit slot at which the complex sum (Eq. (4)) between the delayed versions $u_k(t)$ of the input $u(t)$ is performed. In order to clarify the role of the best sampling, we study in details the 1-bit delayed XOR at a bit-rate of 5 Gbps by sampling each bit time slot with $B_{sa} = 16$ samples (see "Methods" section). In this case, each sample point is separated by 12.5 ps. Therefore, the perceptron processes four signals delayed by 50 ps at different times of the input bit. Figure 6 reports the BER as a function of the sample number or time computed with respect to the start of the bit, for various input intensity (VOA attenuation).

This measure highlights the system memory by shifting, at steps of 12.5 ps from the start of the bit, the time t_S at which the perceptron outputs the predictor, $y(t_S)$. The 1-bit delayed XOR task requires a memory long enough to get information on the previous bit, therefore the perceptron might be able to compute until t_S reaches the maximum system memory of 150 ps (the maximum delay between u_1 and u_4). At this time, all 4 delayed signals carry information on just the current bit as is shown in the upper panel of Fig. 6. The best BER performance is obtained for $t_S = 62.5$ ps. The network trained on the first 3 sampling intervals (i.e., $t_S < 50$ ps) cannot solve the task. In this time frame, the sum (4) is performed on u_1 coming from the actual bit (note that u_1 is the most intense because it is not delayed, i.e., it does not propagate through the spirals) and on u_2, u_3 and u_4 coming from the past bit that are attenuated by the spiral losses. As shown in Fig. 2, the best performance on the XOR task is obtained when u_1 and u_2 are equal since, by adjusting their relative phase, also their amplitude can be modified in the sum due to the interference between signals coming from the same bit. This condition is reached at $t_S >$
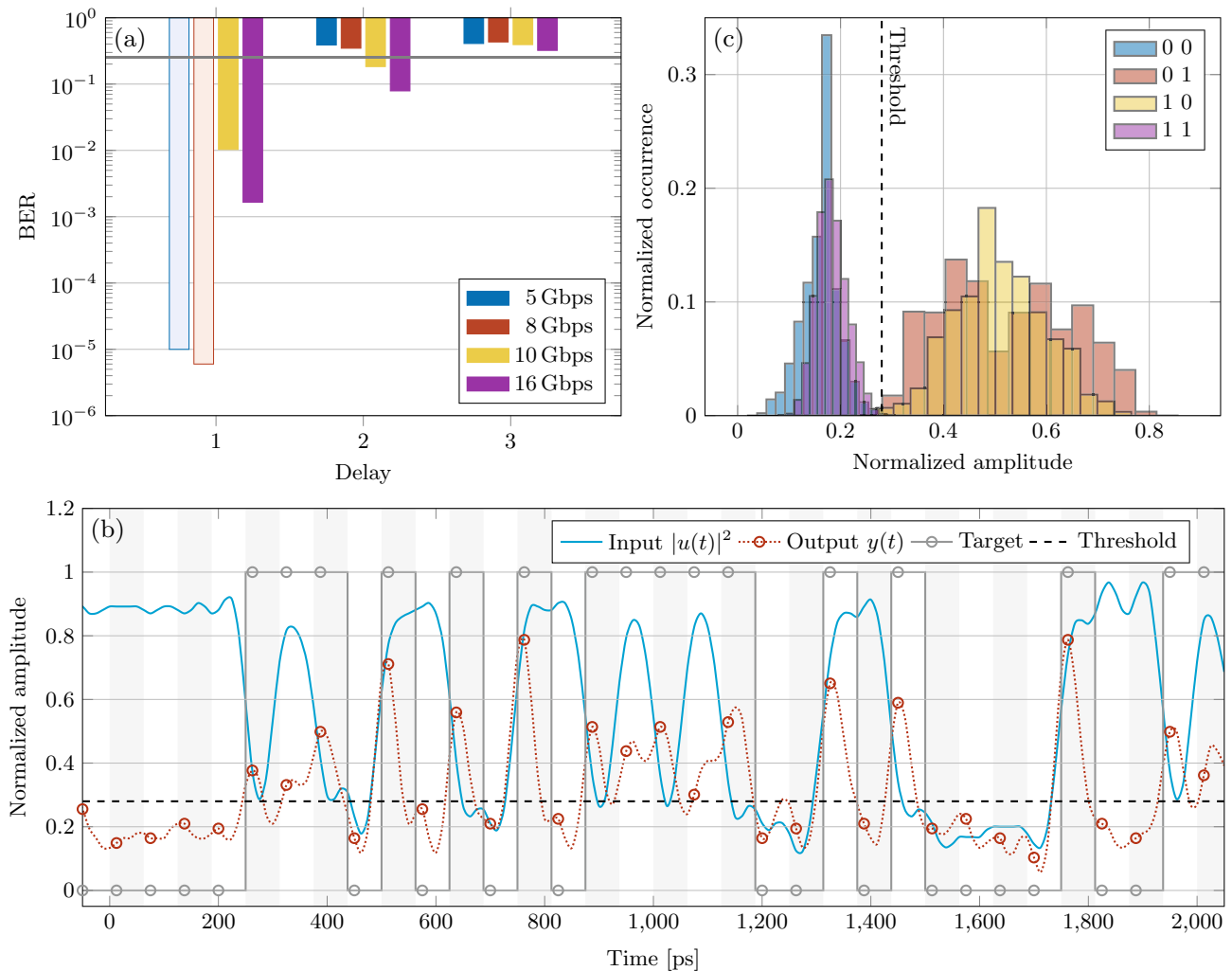
Figure 5. (**a**) BER for the several bit rates indicated in the inset and XOR bit delay. The BER is indicated with transparent bars when the statistical limit is achieved, i.e., the operation is error-free. The black line refers to the non-linear separability threshold. (**b**) Input u (blue) and output y (dashed red) time traces for a bit rate of 16 Gbps. The vertical stripes show the bit slots, the horizontal black dashed line indicates the optimal threshold while the red dots the optimal sampling time. Gray dots mark the target. (**c**) Output signal levels distribution of the 2nd bit of the 2-bit patterns given in the inset. The vertical dashed line marks the optimal threshold. (**b,c**) Refer to a bit rate of 16 Gbps. Training and test are carried out at best sampling and threshold values.

50 ps. The performance of the perceptron degrades when also u_3 is populated by the actual bit and fades away when the perceptron memory is overcome.

This simplified scheme is affected by the setup jitter which influences the exact timings, shuffling the levels used by the perceptron to process the data. Furthermore, a symmetric jitter is expected near the bit end. In this case the scenario is even worse since, as the time shift increases, the information coming from longer spirals have more weight to provide the correct predictor but the bits they convey are attenuated and noisy, thus worsening the overall BER (compared to the first half of the bit).

Figure 6 shows also the effect of the input signal intensity (represented by the VOA attenuation) on the perceptron performance: the higher the input intensity, the better the BER is. The input intensity changes the power at the detector, i.e., its signal-to-noise ratio, but not the perceptron transmission regime that remains linear. This is confirmed by the measured constant insertion loss as the level of the input signal is varied.

We performed also the 2-bit delayed and 3-bit delayed XOR tasks (Fig. 5a). Results show that only for the 2-bit delayed XOR at the highest bit rates (10 and 16 Gbps), the complex perceptron exceeds the non-linear separability threshold (solid horizontal line). This is because only at these rates the two required conditions of equal u_1 and u_2 with the present bit and, at the same time, of u_3 and u_4 having the 2-bit delayed bit can be achieved (indeed for these rates the bit periods are 100 and 62.5 ps, respectively). For the 3-bit delayed XOR task, this condition is never achieved and the perceptron predictions are worse than the non-linear separability threshold. A rate larger than 20 Gbps would be needed to perform this task.
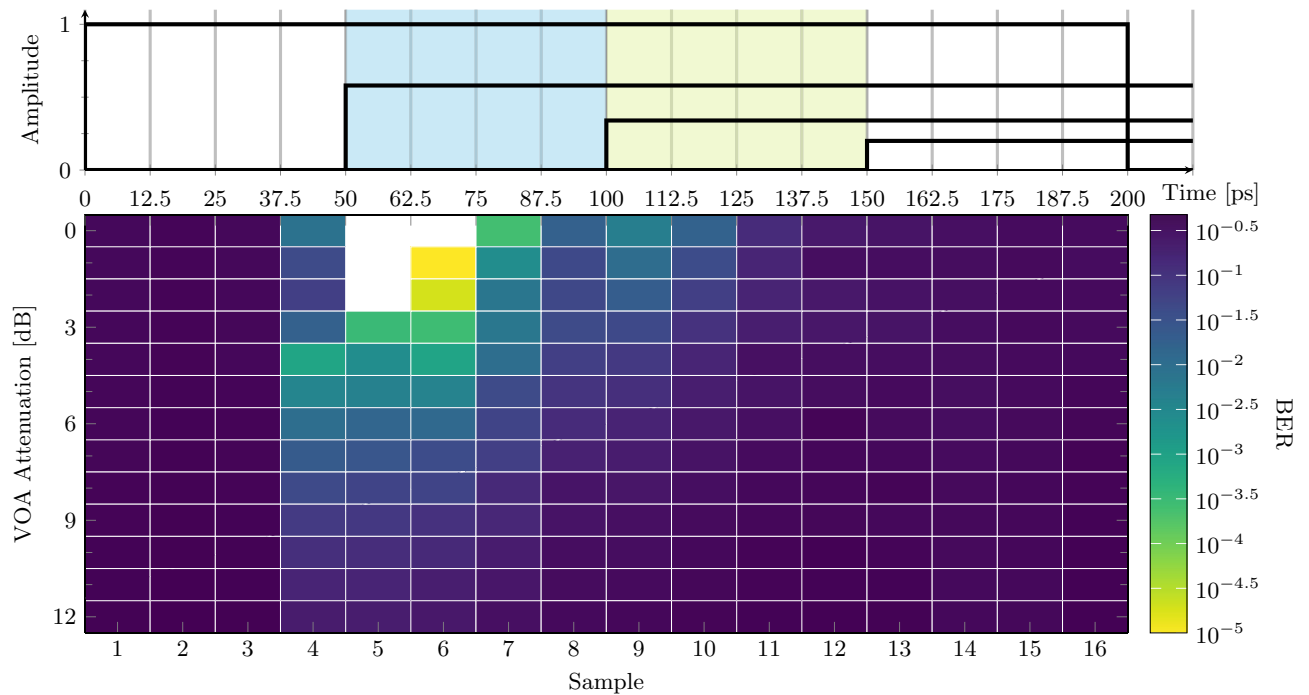
Figure 6. (Upper) Perceptron delayed inputs u_i (black lines) as a function of time from the begin of the bit for an input equal to 01. The delayed copies are attenuated accordingly to the spirals losses. Vertical blue and green stripes identify the optimal working region. (bottom) 1-bit delay XOR at 5 Gbps. BER versus the time from the begin of the bit and VOA attenuation. Data at the best BER: VOA attenuation 0 dB, best sampling time 62.5 ps, insertion loss 9 dB, power at the detector 1.1 dBm, input power 4 dBm, BER statistical limit 10^{-5}.

Complex valued optical perceptron modeling. To further underline the performances of the complex perceptron, we compare a simulation of the perceptron with the experimental data and we benchmarked it with other schemes based on a similar simple optical circuit. Three photonic neural networks have been used:

- The complex-valued perceptron. In this case, the network is modeled as a complex valued perceptron with delayed inputs and $|\cdot|^2$ activation function (Eq. (4)). The training is performed using the PSW algorithm where only phases are trained. This is the model of the measured device.
- The real-valued perceptron. In this case, the network is modeled as a real valued perceptron. Specifically, the modulus square is applied directly on the delayed input copies, i.e., no $|\cdot|^2$ activation function is applied. The delayed copies are then weighted with real numbers and summed to produce the prediction. The training is performed using a ridge regression and the amplitudes of the weights are changed.
- The reservoir computing network with virtual nodes[15]. The complex perceptron is used as a reservoir with random phases. The samples of the output are used as virtual nodes. The network output is then computed as the weighted sum of the virtual node states, where the optimal weights are found through ridge regression. To simulate the random connections of the reservoir, the perceptron output has been determined by giving 4 random currents to the heaters 10 times. The performance is then calculated as the average and best results over the repetitions.

The simulated performances of the three networks on the 1-bit delayed XOR task at 5 Gbps and 4 dBm input power are reported in Fig. 7. All networks are trained and tested with B_{sa} = 16. For the reservoir computing network this is the number of virtual nodes. The experimental data (blue line) is extracted from Fig. 6 at 2 dB VOA attenuation.

The model of the complex-valued perceptron (red line) reproduces the experimental data (blue line) when a phase noise of 1% is added. Phase noise accounts for any normally distributed fluctuation of the weights around the trained values. When the value of propagation losses is lowered to the more reasonable value of 2.5 dB/cm (yellow line), the performance at longer time is improved. The real-valued perceptron (violet line) does not solve the 1-bit delayed XOR due to the lack of the non-linearity. The average performance of the reservoir (dashed line) is not enough to solve the XOR task. Instead, the best case scenario (solid line) solves the task, but at a BER that is order of magnitude worse than the result of the complex perceptron.

Phase encoding recognition. A further interesting characteristics of the complex perceptron is the ability to handle pure phase information. In fact, since the perceptron is based on phase modulation only, it is able to decode phase encoded information, as well. This is a relevant task, which can be used in coherent detection or in protocols of secure communication. Figure 8 reports the results for the trained perceptron which is instructed
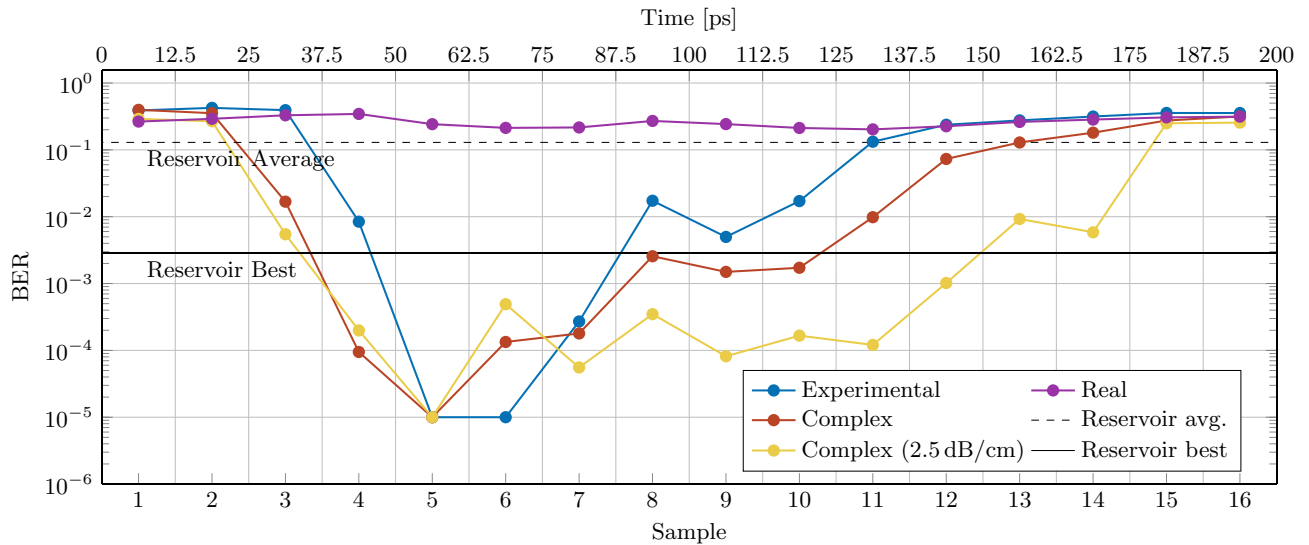
Figure 7. 1-bit delayed XOR. BER as a function of the time from the begin of the bit for the different models indicated in the inset. Simulation parameters: input power 4 dBm, bit-rate 5 Gbps, BER statistical limit 10^{-5}.
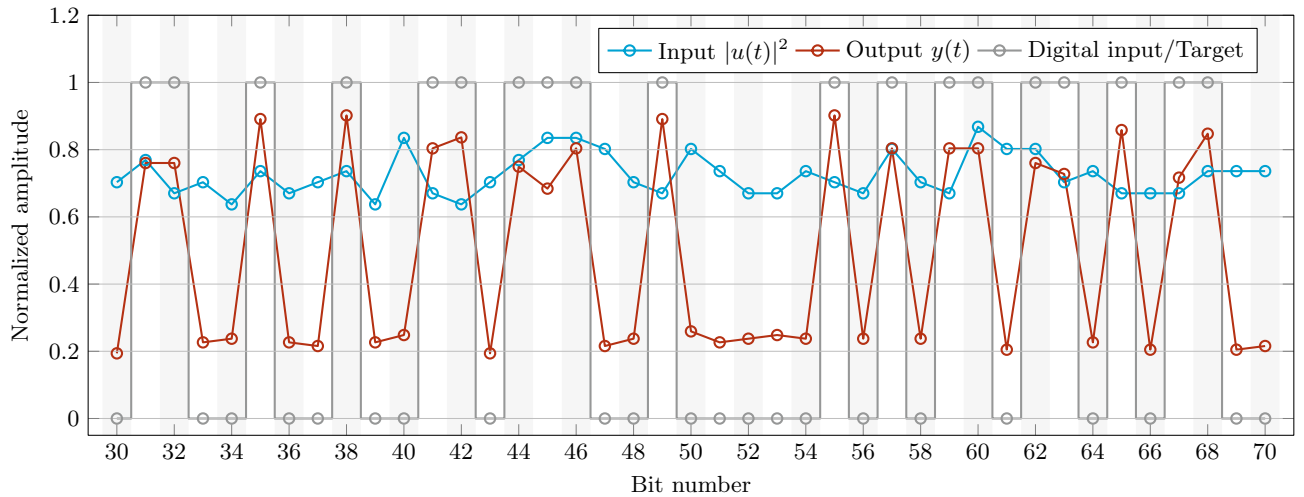

Figure 8. Input signal intensity (blue), output signal intensity (red) and digital input phase (gray) as a function of the bit number.

to translate the phase encoded modulation of the input bit sequence to amplitude modulation of the output sequence.

Gray circles show the input bits encoded on the input signal phase (bit 1 phase π, bit 0 phase 0), red circles show the the signal output. Decoding is done with zero errors on a 10 Gbps sequence. The Pearson correlation coefficient between the input intensity and the trained output is 0.002, which proves that the trained output is not determined by the input signal intensity. Remarkably, the input signal (blue circles) measured on a detector shows only amplitude fluctuations due to noise that are not correlated with the phase encoded information.

Discussion

We demonstrate a silicon photonic integrated optical perceptron based on a multiple delayed interferometer that performs logical operation and pattern recognition up to 16 Gbps (limited by our testing system). The experimental results reflect the complicate interplay between the input bits sequence, the delay lines, and the non-linearly modulated interference that outputs the prediction. For example, the device is expected to perform at its best around a bit duration that is close to the delay of the spiral, i.e., around 16 Gbps. On the contrary, higher performance is reported at low bit-rates for all the tasks where 1 bit of memory is required (2-bit case). This fact is related to the training method in which the best sampling time is a free parameter chosen during the training session.

In fact, when the bit duration exceeds the device memory, the best sampling shifts towards the transient between the past and current bits to retain enough memory to solve the tasks while minimizing the jitter. On
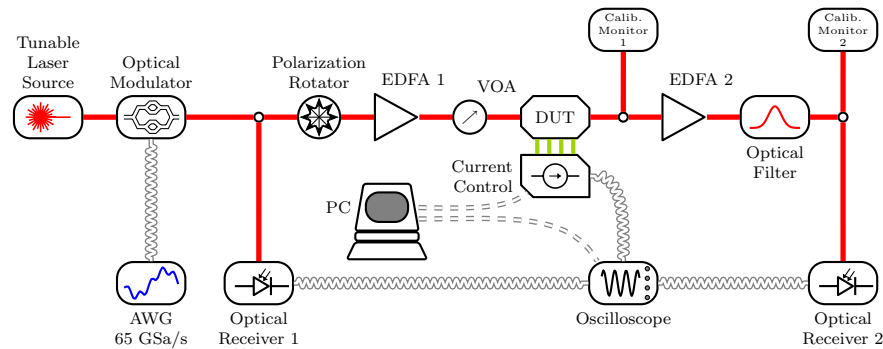
Figure 9. Experimental setup. The different components are labeled in the figure and explained in the text. The complex perceptron is indicated as the DUT (Device Under Test). Red lines refer to optical signals, black lines to electrical signals.

the 3-bit case, its best performance is obtained at the highest bit rates and depends on the exact input sequence, since a 3-bit pattern uses its full memory.

It is worth noting that, once trained, the complex perceptron is a passive device that, on the chip, consumes only the electrical power to run the various phase shifters. In these, 30 mW are dissipated. This value can be lowered by optimizing the phase shifter efficiency, i.e. by using trenches, or by implementing other schemes to control the phase, such as the plasma dispersion effect.

The off-the-chip activation function, required to get to the output as in Eq. (3), can be easily integrated by means of integrated, high speed photodiode without degrading system performance. This is a component already offered as a standard block by most of the silicon photonics foundries.

The total optical losses depend on the fiber-chip coupling system, the type of solved task and the delay lines efficiency. For the XOR task, which is the most demanding one, we measured a total insertion loss of 9 dB and, given coupling losses of 7.6 dB, only 2.4 dB are the device losses.

We demonstrate that the complex perceptron can solve several binary tasks by training all system parameters based on phase, as in FFN, thus exploiting the full PIC resources. It is also able to compare samples from nearby bits to unveil temporal correlations, as in a RC schemes. Unlike the systems reported in recent works[21], the system computes in the analog domain, the digital-to-analog conversion is carried out only at the perceptron output to read the computed data.

The system memory provided by the spirals is linear, thus it can be scaled to higher level assuming that losses can be significantly reduced. Differently from the nonlinear RC schemes[22], the memory provided by the spiral is limited only by the amount of propagation losses and the receiver sensitivity. An estimate of the maximum memory obtainable without amplification in terms of the number of bits N_{bit} stored in the perceptron, can be obtained from:

$$N_{bit} = \frac{a_t\, n_g}{c\, T_{bit}\, \alpha_{pl}}, \tag{5}$$

where a_t is the maximum admitted loss, n_g the group index, c the speed of light, T_{bit} the bit duration as the inverse of the bit-rate and α_{pl} the propagation loss. By considering an ideal α_{pl} = 2dB/cm, n_g = 4.2 as for the used waveguides, T_{bit} = 33 ps (i.e. 30 GHz, limited by commercially available AWG) and by fixing a_t =10 dB, we get a memory of 21 bits. Note that a_t considers the losses due to the other components in the perceptron as well, and not only the propagation losses in the spirals.

The perceptron is trained changing only the phases of complex-valued weights, which demonstrates the possibility to avoid direct losses caused by the amplitude variations. A proper design of the node topology and interconnections will permit to compare samples from bits distant in time and to expand the capability of the network to correlate such information.

Methods

Experimental setup. The experimental apparatus is sketched in Fig. 9. The laser source is a C-band, CW tunable laser (Pure Photonics) modulated by an electro-optic IQ modulator (IxBlue MXIQ-LN-30) to create the desired input waveform. The modulator is driven by a 65 GSa Arbitrary Waveform Generator (AWG) from Keysight (KS8195A), whose output is amplified by a high bandwidth amplification stage (IxBlue DR-AN-28-MO), providing the necessary voltage swing of V_π = 7V to exploit the full dynamic range of the modulator. A tap of 10% is placed at the output of the modulator to monitor the input pump, which is detected by a fast photodiode (RX1, Thorlabs DXM20AF, 20 GHz bandwidth). The polarization is rotated to match the TE required by the device. The pump is amplified to a fixed level of 20 dBm by an Erbium Doped Optical Amplifier (EDFA, IPG Photonics) and the input power is regulated by an electronic Variable Optical Attenuation (VOA, VIAVI mVOA-C1). As a consequence, this VOA also regulates the power at output photodiode (RX2, Thorlabs DXM20AF) that ranges between 4 and 10 dBm. The power level at the device output is monitored through a tap of 1% and a low noise photodiode (M1, Viavi mOPM). Light is amplified by a second EDFA (Thorlabs EDFA100s) operating at con-

stant current. A tunable optical band-pass filter (25 GHz bandwidth) removes the broadband ASE noise prior to reach RX2. Another photodiode (M2, mOPM) monitors precisely the power reaching RX2. A 2×80 GSa/s oscilloscope with 16 GHz analog bandwidth (LeCroy SDA 816Zi-A with interleavers) records the input $u_{in}(t)$ and the output $u_{out}(t)$ waveforms. A computer controls the current flowing in the 4 heaters integrated onto the device through an 8 channels current generator (Qontrol Q8iv). An extra current output is sent to the trigger input port of the oscilloscope as time reference. The training algorithm runs on the PC that modulates the currents accordingly to the acquired input and output waveforms.

Integrated device. The device has been fabricated on a CMOS facility on silicon-on-insulator wafer with a device layer thickness of 220 nm (iSiPP50G technology process by IMEC through a MPW scheme). Silicon waveguides (WG) are embedded in a silica cladding, and have a width of 450 nm to ensure single mode operation on both polarizations. TiN tracks deposited on top of the silica cladding enables local thermal tuning of WG effective index. A CMOS packaging holds and thermalize at 21 °C the photonic chip using a PID controller and a Peltier heater. The measured single mode WG propagation loss measured through spirals of different lengths is $\sim$ 6 dB/cm. This value is highly above the 2 dB/cm average performance of IMEC iSiPP50G process. Therefore, we consider the device performances to be sub-optimal. Fiber-to-chip coupling is ensured by grating couplers whose measured insertion loss is $\sim$ 3.8 dB/grating at the maximum transmission wavelength of 1560 nm.

Perceptron training and testing procedures. We used a Particle Swarm (PSW) algorithm to train the optical node[19]. Despite its stochastic nature, the limited number of parameters of our system makes it the easiest way to optimize our perceptron as it does not require direct access to the fields in the different waveguides and the activation function can be considered without any approximation. More complex networks would benefit of gradient-descent, back-propagation-like algorithms, which assure faster convergence and have been reported for photonic implementation of FFN in Ref.[11].

The training is carried out by a PC that elaborates the output signal and regulates the current control following the PSW algorithm, as shown in Fig. 9. The typical training time is few tens of seconds and is mainly limited by oscilloscope-PC data exchange. An FPGA/ASIC implementation of the PSW would greatly decrease this time, which then would become limited only by the speed of the mechanism used to change the weights (MHz in our system). The input signal is a 8-bit Pseudo Random Binary Sequence (PRBS) that is amplitude modulated NRZ (not return to zero) between 5 and 16 Gbps, with an extinction ratio and a SNR of 7 dB and 14 dB, respectively. The delay between the input (RX1) and output trace (RX2) is found by cross correlating the traces. This operation is performed every time the bit rate of the experiment is changed. The operations of the training phase, performed at each PSW iteration, are:

- Current values are generated by the PSW and are applied by the current controller to the heaters while the trigger signal is sent to the oscilloscope. At this time, only 3 phases are trained to avoid redundancy in the loss function due to the system intrinsic periodicity of 2π.
- To take into account the dynamics of the heater controllers, we use a delay of 1 ms after the trigger signal before the oscilloscope acquires the input (x_j^{RX1}) and output (x_j^{RX2}) traces sampled at 80 GSa/s for a total of M = 160 kSa, j = 1,..., M. The number of processed bits depends on the bit rate, for instance, at 16 Gbps there are a total of 10^4 bit and 5 Samples/bit.
- The PC aligns the traces using the already measured delay for that specific bit rate.
- x_j^{RX1} is digitized (X_l^{RX1}) using its average as threshold and selecting the central bit sample. The bit index l goes from 1 to M/B_{Sa}, where $B_{Sa} = \frac{80 \times 10^9}{\text{bit rate}}$ is the number of samples per bit. The target binary sequence T_l is then calculated by applying the task function to X_l^{RX1}.
- x_j^{RX2} is digitized (X_l^{RX2}) using a variable threshold $r = w_{min}^{RX2}, ..., w_{max}^{RX2}$ spanning the signal dynamic range of the samples $n = 1, ..., B_{sa}$.
- The loss function is calculated as $L_{rn} = \sum_{l=1}^{M/B_{sa}} |X_l^{RX2} - T_l|^2$. The minimum of L_{rn} is considered by selecting the best threshold r_b and the best sampling position n. The BER is calculated as $BER = \frac{L_{rn}}{MB_{Sa}}$.

The algorithm iterates these steps either until error free or the max iteration number are reached. Note that each input trace provided to the perceptron during the training is unique since the various traces are all corrupted by the experimental noise (such as the detector thermal and shot noise, the phase noise on the weights due to the micro-thermal variation on the spirals, and the time jitters of the AWG/oscilloscope sampling times). Such noise slows down the training process but helps in avoiding over-fitting. As a result of the train the 3 optimal currents found are used in the testing phase where 10 new traces are acquired and processed. The test BER is the average of the BER calculated for each acquisition.

References

1. Xu, X. *et al.* 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589**, 44 (2021).
2. Feldmann, J. *et al.* Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589**, 52 (2021).
3. Vandoorne, K., Dambre, J., Verstraeten, D., Schrauwen, B. & Bienstman, P. Parallel reservoir computing using optical amplifiers. *IEEE Trans. Neural Netw.* **22**, 1469 (2011).
4. Teğin, U., Yıldırım, M., Oğuz, İ, Moser, C. & Psaltis, D. Scalable optical learning operator. *Nat. Comput. Sci.* **1**, 542 (2021).

5. Katumba, A., Yin, X., Dambre, J. & Bienstman, P. A neuromorphic silicon photonics nonlinear equalizer for optical communications with intensity modulation and direct detection. *J. Lightwave Technol.* **37**, 2232 (2019).
6. Argyris, A., Bueno, J. & Fischer, I. Photonic machine learning implementation for signal recovery in optical communications. *Sci. Rep.* **8**, 1 (2018).
7. Harris, N. C. *et al.* Linear programmable nanophotonic processors. *Optica* **5**, 1623 (2018).
8. Peng, H.-T., Nahmias, M. A., De Lima, T. F., Tait, A. N. & Shastri, B. J. Neuromorphic photonic integrated circuits. *IEEE J. Sel. Top. Quantum Electron.* **24**, 1 (2018).
9. Genty, G. *et al.* Machine learning and applications in ultrafast photonics. *Nat. Photonics* **15**, 91 (2020).
10. Shastri, B. J. *et al.* Photonics for artificial intelligence and neuromorphic computing. *Nat. Photonics* **15**, 102 (2021).
11. Hughes, T. W., Minkov, M., Shi, Y. & Fan, S. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* **5**, 864 (2018).
12. Xu, X. *et al.* Photonic perceptron based on a Kerr microcomb for high-speed, scalable, optical neural networks. *Laser Photonics Rev.* **14**, 2000070 (2020).
13. Carroll, T. L. Do reservoir computers work best at the edge of chaos? *Chaos Interdiscipl. J. Nonlinear Sci.* **30**, 121109 (2020).
14. Appeltant, L. *et al.* Information processing using a single dynamical node as complex system. *Nat. Commun.* **2**, 1 (2011).
15. Duport, F., Schneider, B., Smerieri, A., Haelterman, M. & Massar, S. All-optical reservoir computing. *Opt. Express* **20**, 22783 (2012).
16. Vandoorne, K. *et al.* Experimental demonstration of reservoir computing on a silicon photonics chip. *Nat. Commun.* **5**, 1 (2014).
17. Tait, A. N., Nahmias, M. A., Shastri, B. J. & Prucnal, P. R. Broadcast and weight: An integrated network for scalable photonic spike processing. *J. Lightwave Technol.* **32**, 4029 (2014).
18. Rosenblatt, F. *The Perceptron, a Perceiving and Recognizing Automaton Project Para* (Cornell Aeronautical Laboratory, 1957).
19. Bansal, J. C., Singh, P. K. & Pal, N. R. *Evolutionary and Swarm Intelligence Algorithms* (Springer, 2019).
20. Schubert, F. & Gros, C. Local homeostatic regulation of the spectral radius of echo-state networks. *Front. Comput. Neurosci.* **15**, 12 (2021).
21. Shen, Y. *et al.* Deep learning with coherent nanophotonic circuits. *Nat. Photonics* **11**, 441 (2017).
22. Inubushi, M. & Yoshimura, K. Reservoir computing beyond memory-nonlinearity trade-off. *Sci. Rep.* **7**, 1 (2017).

Acknowledgements

Author contributions

M.M. conceived the experiment. M.M. and D.B. designed the device, made the experiments and analyzed the data. M.M. simulated the experiment. P.B. and L.P. supervised the project. All authors contributed to the discussion of the results and to the writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.M.

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.