Learning in Photonic Neural Networks

Giacomo Antonioli 619292
Computational Neuroscience Seminar

1.

What are Photonic Neural Networks?

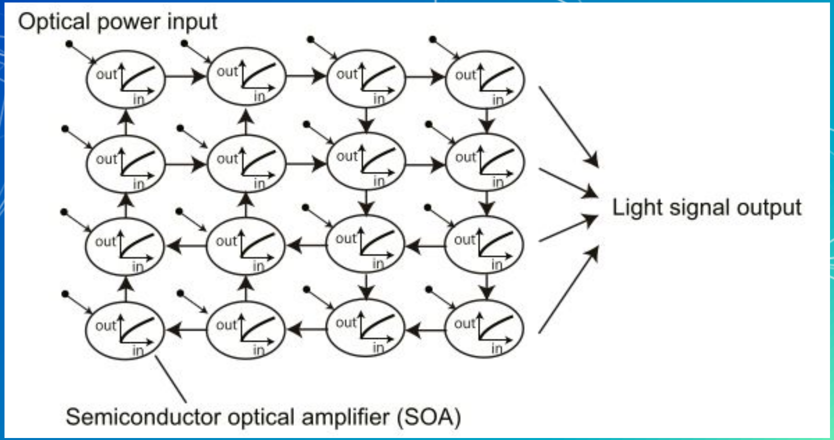
Photonic Neural Networks

- Photonic NN are a physical implementation of Neural Network models that use optical hardware to implement the Net structure.

2.
How are they structured?

Photonic Neural Networks[1]

- Many different physical and exotic implementations were tried.
- Some implementations use chip-integrated semiconductors assembled in arrays connected to a maximum of **n** neighbors. These can be defined as optical arrays implementations.



Photonic Neural Networks

- Other implementations are based on optical fibers, photodiodes, and modulators. These implementations are defined as Time-delayed Photonic RC's.

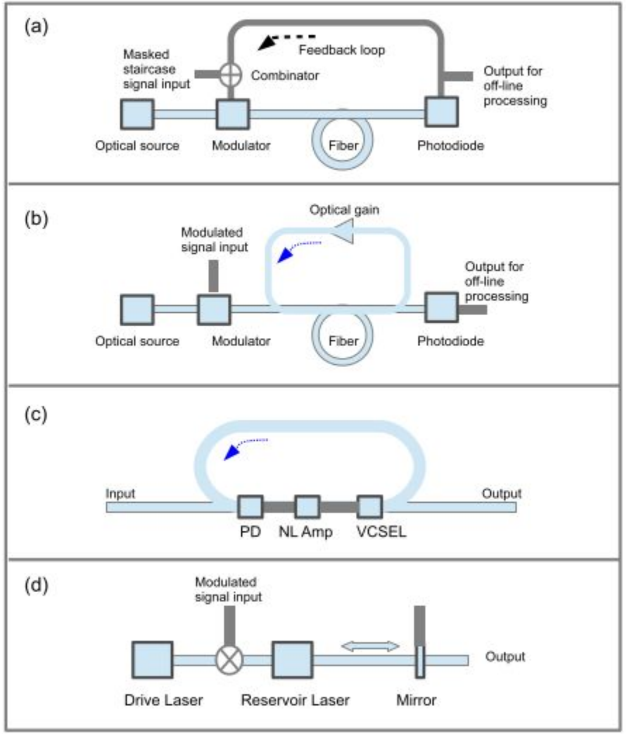
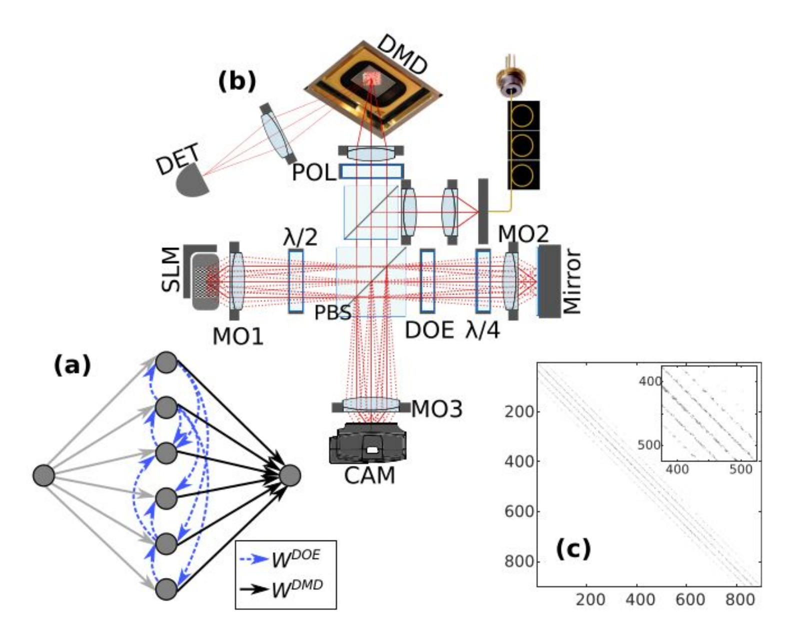


Fig. 9. Time-delay reservoir configuration examples. Blue and gray lines represent optical and electronic signals, respectively. (a) Opto-electronic feedback loop. (b) All-optical feedback loop with gain. (c) Passive optical feedback and opto-electronic node with gain. (d) Feedback into a laser cavity by a partially reflecting mirror . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Photonic Neural Networks

One recent implementation was made by An optical plane wave that illuminates the spatial light modulator (**SLM**), the neural network states are encoded by the SLM pixels. These are imaged on the camera, passing through a polarizing beam splitter (**PBS**) and the diffractive optical element (**DOE**) creating the coupling between network states. The information detected by the camera is used to drive the SLM.

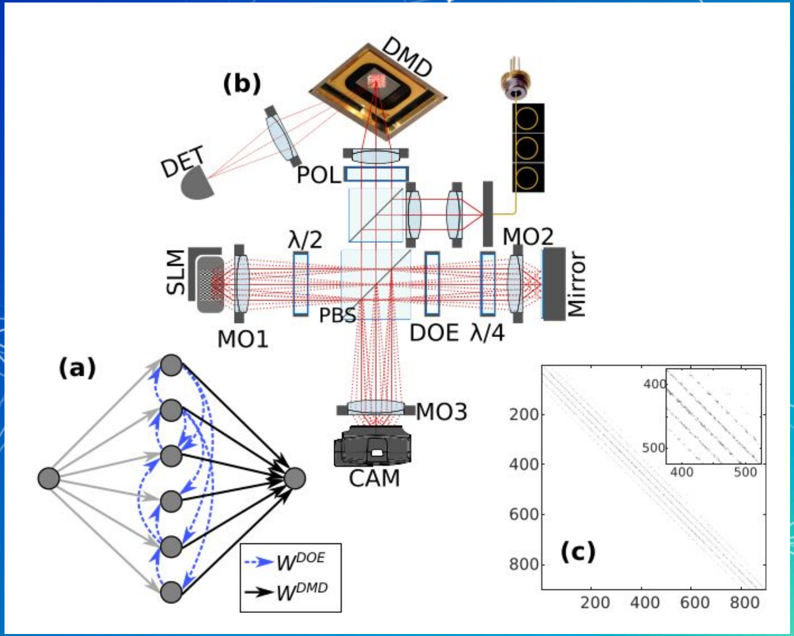


The network's output weights are realized via a digital micro-mirrors device (**DMD**) which creates Boolean readout weight matrix $\mathbf{W}^{\mathbf{DMD}}$.

3.
Why physical implementation?

Photonic Neural Networks

- Physical implementations have many gains: they are more efficient and can be more power efficient to. Once the net is trained only power needed is the one to interpret the optical output signal.
- Also they can be implemented independently as stand-alone machines.



4.
How do they work and how do they learn?

Photonic Neural Networks

- Photonic Neural Networks are still in a development phase, impling no main direction of development. Many different approaches have been taken both in the architectural approach and the learning approach.

Spiking Neural Networks (SNN)[2]

- All-optical **SNN** is a network that consists of optical spiking neurons. The architecture of the proposed all-optical SNN based on VCSELs* .In an all-optical SNN, the input and output signals are expressed as spikes.

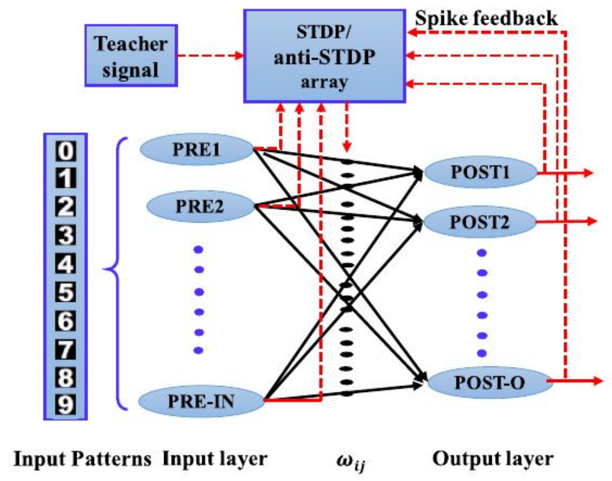


Fig. 1. Architecture of the proposed all-optical SNN. Spatiotemporal inputs and spatiotemporal outputs are considered; 0–9 are ten input patterns, PRE-1–PRE-IN represent the PREs based on VCSEL-SA, POST-1–POST-O denote the POSTs based on VCSEL-SA, and teacher signal is the target spikes. The training is performed by the ReSuMe method based on the STDP and anti-STDP rules.

***V**ertical **C**avity **S**urface-**E**mitting **L**aser

Spiking Neural Networks (SNN)

- During each training epoch, the weight update is determined by the timing of the presynaptic spikes, postsynaptic spikes, and the supervisory spikes. The purpose of supervised learning is to make the POSTs emit at desired target timings.

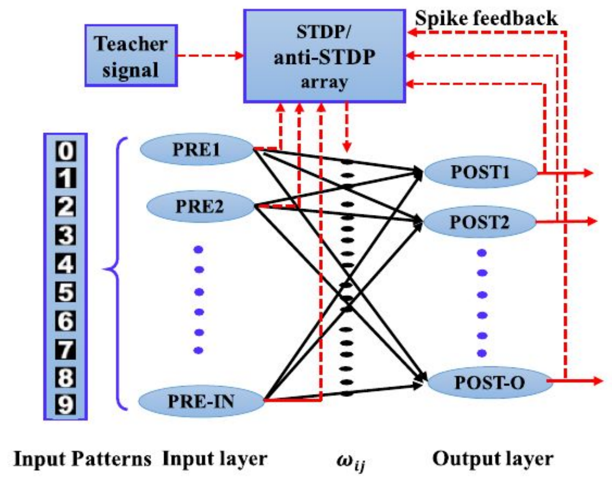


Fig. 1. Architecture of the proposed all-optical SNN. Spatiotemporal inputs and spatiotemporal outputs are considered; 0–9 are ten input patterns, PRE-1–PRE-IN represent the PREs based on VCSEL-SA, POST-1–POST-O denote the POSTs based on VCSEL-SA, and teacher signal is the target spikes. The training is performed by the ReSuMe method based on the STDP and anti-STDP rules.

Spiking Neural Networks (SNN)

- The training process is implemented by the STDP-based ReSuMe supervised learning method*.

$$\Delta\omega_i = \omega_f \times \Big[(m_d - m_o) + \sum_{t_d}\sum_{t_i \le t_d} \Delta\omega_{\mathrm{STDP}}(t_d - t_i) + \sum_{t_o}\sum_{t_i \le t_o} \Delta\omega_{a\mathrm{STDP}}(t_o - t_i)\Big], \quad \text{if } |t_o - t_d| > r \tag{12}$$

$$\omega_i(x+1) = \omega_i(x) + \omega_f \times \Delta\omega_i. \tag{13}$$

* F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," Neural Comput., vol. 22, no. 2, pp. 467–510, Feb. 2010

Spiking Neural Networks (SNN)

- ωi represents the weight update amount. Two processes are an STDP process for potentiating synapses based on input spike trains and desired target spikes and an anti-STDP process for depressing synapses based on the input spike trains and actual output spikes.

- For a pattern recognition task, we assume that the recognition is successful when |to – td | < r is satisfied, where to (td) denotes the output (target) spike timing. In the tests made by the authors the accepted interval was 0,5s.

$$\Delta\omega_i = \omega_f \times \Big[(m_d - m_o) + \sum_{t_d} \sum_{t_i \le t_d} \Delta\omega_{\mathrm{STDP}}(t_d - t_i) + \sum_{t_o} \sum_{t_i \le t_o} \Delta\omega_{a\mathrm{STDP}}(t_o - t_i) \Big], \quad \text{if } |t_o - t_d| > r \tag{12}$$

$$\omega_i(x+1) = \omega_i(x) + \omega_f \times \Delta\omega_i . \tag{13}$$

Spiking Neural Networks (SNN)

- A further step in obtaining a more network comparable with the human visual system two prior determined fixed convolutional filters were introduced: one with the kernel size of 7 × 7 and the other kernel is of size 5 × 5.

- This was also done to reduce the possible noise introduced in the input of the net.

- With low noise both implementations reached an accuracy rate of 1. Progressively introducing higher levels of noise showed how using convolutions made the system more resilient.

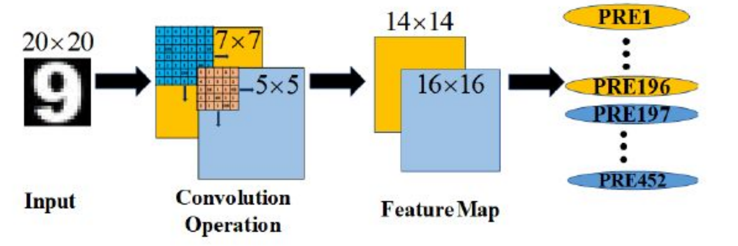
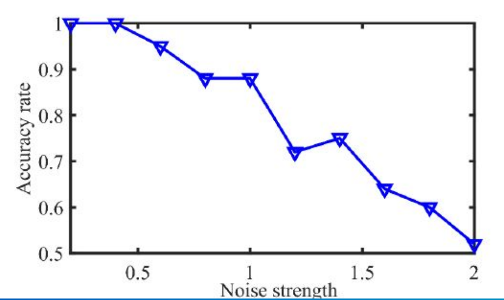
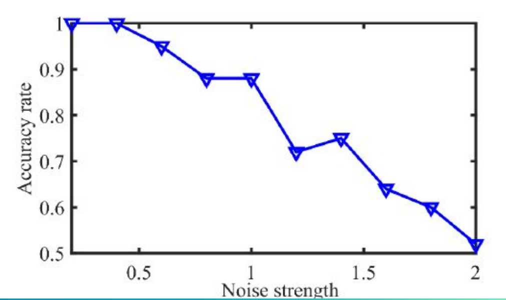


Fig. 17. Schematic of image preprocessing with the convolution operation in the spike encoding phase.

SNN with no convolutions



SNN with convolutions



Reservoir Computing [3]

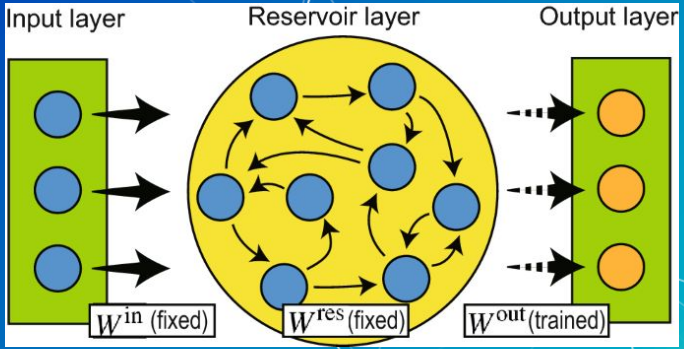
- Before introducing the next implementation what a Reservoir Computing Network is has to be introduced:
  - Reservoir for mapping inputs into high-dimensional space and a readout for pattern analysis. The readout is trained with a simple classificator or regressor.

Reservoir Computing

- The reservoir is made of recurrent neurons and it's weights are randomly initialized respecting the **Echo State Property** (necessary condition) whereby it asymptotically eliminates information for initial conditions.

$$\rho(\mathbf{W}) = \max(abs(\mathrm{eig}(\mathbf{W}))) < 1$$

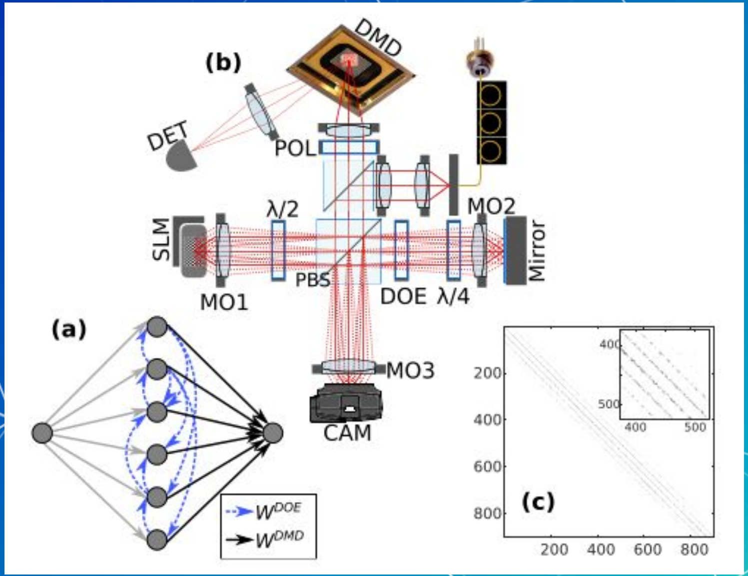
- The only trained part is the readout, so it's faster than other nets.



Reinforcement Learning [4]

- Reinforcement Learning through RC nets was also tried as an alternative way of proceeding. The only part to train was the readout matrix.
- In this model the readout neurons were implemented with Micro mirrors that can be flipped between ±12 °, such that for -12 ° the optical signal is directed to a detector. The detectors photo curent then corresponds to the RNN output.
- With W^{DMD} as the readout weight vector, the RNN output becomes

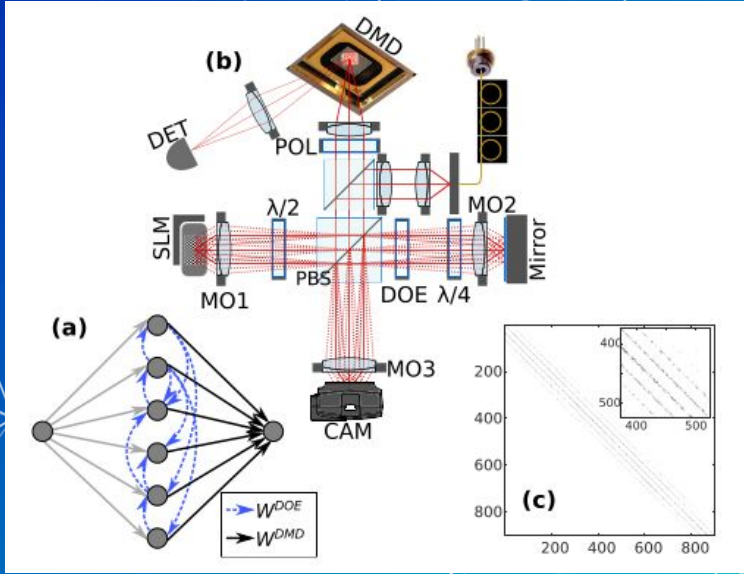
$$y^{out}(n+1) = \delta\left(\mathbf{W}^{DMD}(1 - \mathbf{x}(n+1))\right).$$



Reinforcement Learning

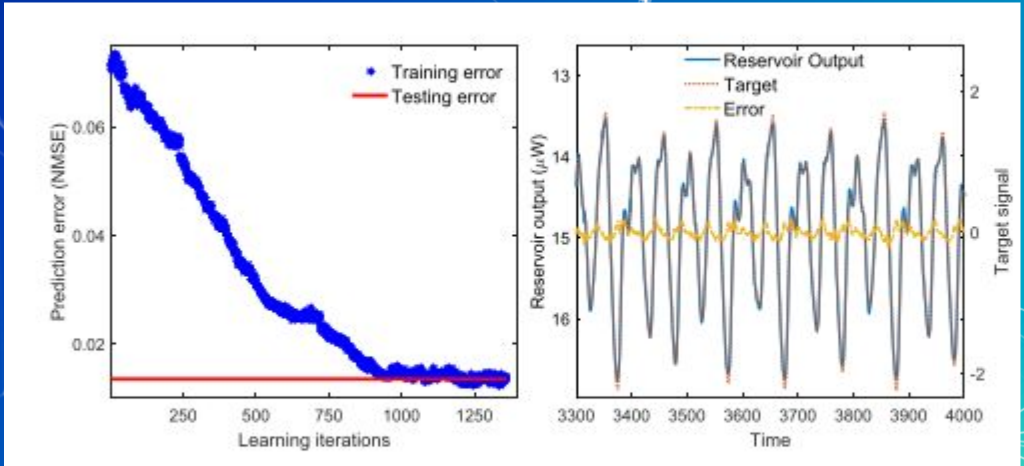
- In this implementation the learning input signal is injected after inverting the weight assigned to one node. The error ε k of signal y_k^{out} (n + 1) obtained for configuration W_k^{DMD} is then compare to the error ε_{k-1} , where k is the index of learning iterations. If the error is reduced, the DMD configuration W_k^{DMD} is kept, if not, it's reverted back to W_{k-1}^{DMD} and a different weight is inverted.
- The weight to be updated is determined by the largest entries $W_k^{select,max}$ position l_k according to:

$$
\begin{aligned}
\mathbf{W}_k^{select} &= rand(N)\cdot \mathbf{W}^{bias},\\
\left[l_k, W_k^{select,max}\right] &= max(\mathbf{W}_k^{select}),\\
W_{k,l_k}^{DMD} &= \neg(W_{k-1,l_k}^{DMD}),\\
\mathbf{W}^{bias} = \frac{1}{N} + \mathbf{W}^{bias}, &\quad W_{l_k}^{bias} = 0.
\end{aligned}
$$



Reinforcement Learning

- For determining the error ε_k the first 30 data points were discarded due to their transient nature. The RNNs remaining output sequence was then inverted, its offset subtracted and normalized by its standard deviation, creating signal $\tilde{y}$ k out .
- The error was measured by $\varepsilon_k = \sigma(y^T - \tilde{y}_k^{out})$, where σ is the standard deviation and ε_k therefore corresponds to the normalized mean square error (NMSE).
- In this implementation all connection weights are positive, and W^{DMD} is boolean. This restricts the functional space available for approximating the targeted input-output transformation.
- To introduce non-linearity it was used the non-monotonous slope of $\cos^2$.



DPNN
Deep Photonic Neural Networks [5]

- One application that developed even more Photonic Networks research was Quantum computing. In particular a research by Google to find a faster way of giving results on user queries*.
- An **ONN** (Optical Neural Network) is a physical implementation of an ANN with optical components. Photons have significantly more computational bandwidth than electrons (limited by clock speed of the circuit), thus can process more data at faster speed.
- Linear transformations (and certain nonlinear transformations) can be performed at the speed of light and detected at rates exceeding 100 GHz in photonic networks, and in some cases, with minimal power consumption.

*Google Search by Quantum Computation Method
https://ieeexplore.ieee.org/document/8385596

DPNN
Deep Photonic Neural Networks

- The issue that a DPNN originally tried to solve was to guide a photon through the waveguides to the desidred counter. Without the DPNN implementations there was no control over the photon pathing.

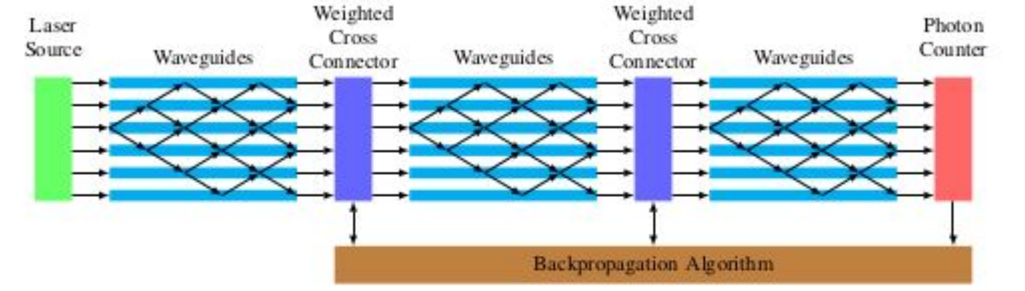


Fig. 1: Deep Photonic Neural Network

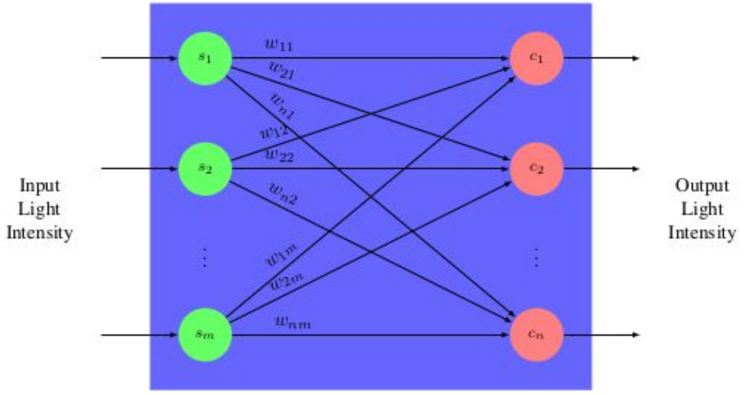


Fig. 2: Weighted Cross Connector

- The photon movement is now controlled by the weighted cross connector

DPNN
Deep Photonic Neural Networks

- Each layer in the represents the photonic lattice and the node in topological diagram represents the waveguide of photonic lattice.
- The cross connections in the topological diagram are represented by weighted cross connector shown in the.

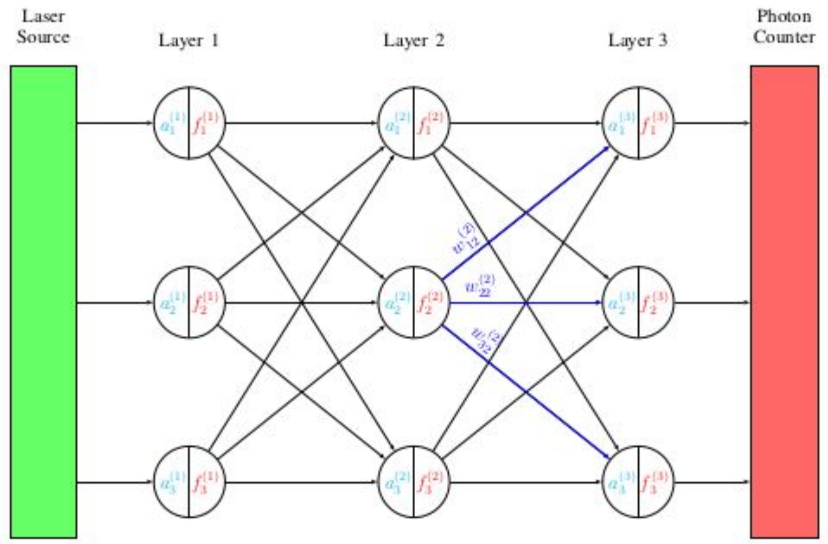


Fig. 3: Deep Photonic Neural Network Topology

DPNN
Deep Photonic Neural Networks

- The procedure of updating the weights uses backpropagation and gradient descent.
- Hardware implementation of ANNs imposes a variety of constraints:
- Weights have to be ≥ 0. This is because negation of light intensity is not possible.
- The sum of the weights has to be 1. This is because light intensity is divided to different waveguides.
- This two constraint can be satisfied applying a Sigmoid function and then a Softmax function.

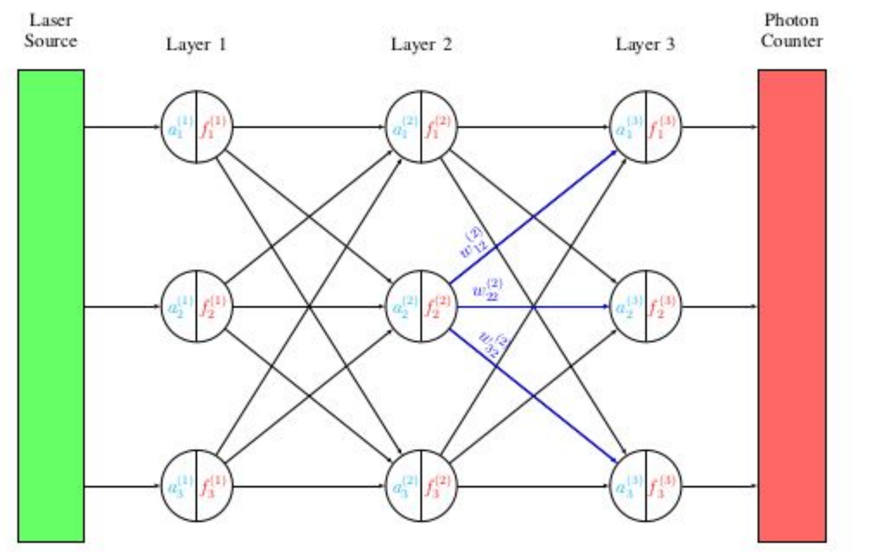


Fig. 3: Deep Photonic Neural Network Topology

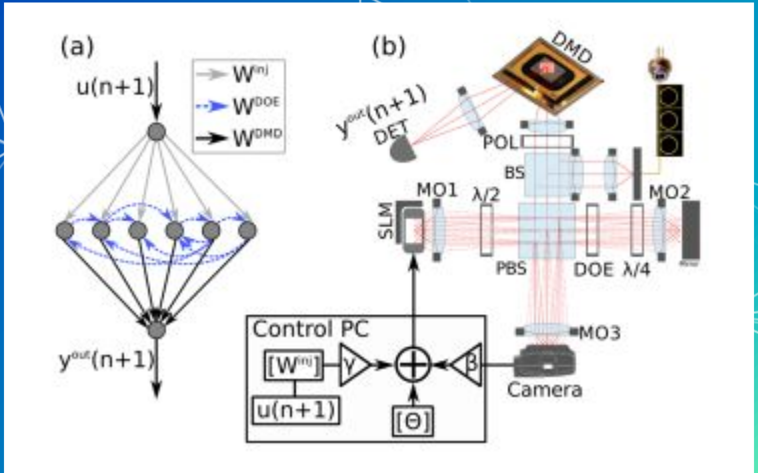
5.
Does the physical implementation introduce noise?

Photonic Neural Networks [6]

- Certainly having a Physical implementation implies different types of noise due to a non ideal system.
- Lens and optical circuits are not noise-proof. Diffraction cause by the used material has to be taken in account during the Learning Phase.

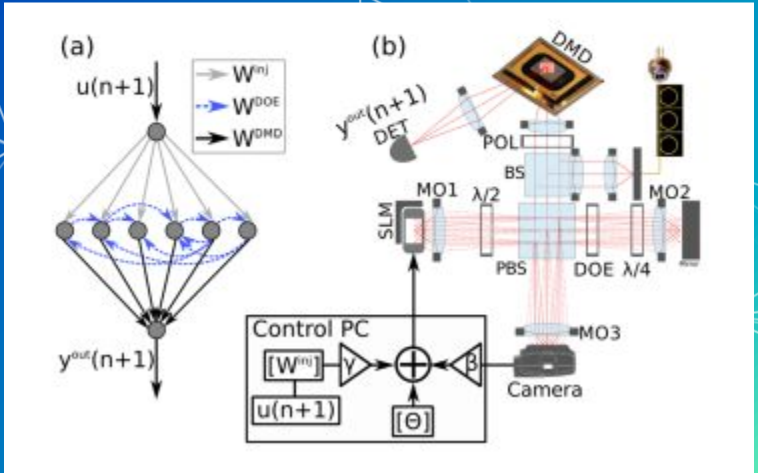
Noise under Boolean Learning

- Some experiments have been made to evaluate how noise affects learning and learning rules. To do this a network with DMD's and Boolean learning has been developed to study the effects of noise.
- The Network implemented was RC network and as usual only the readout layer was trained.
- The physical constraint ar the same as in the Reinforcement Learning Network that is positive weights and $\cos^2$ as non linear activation function.



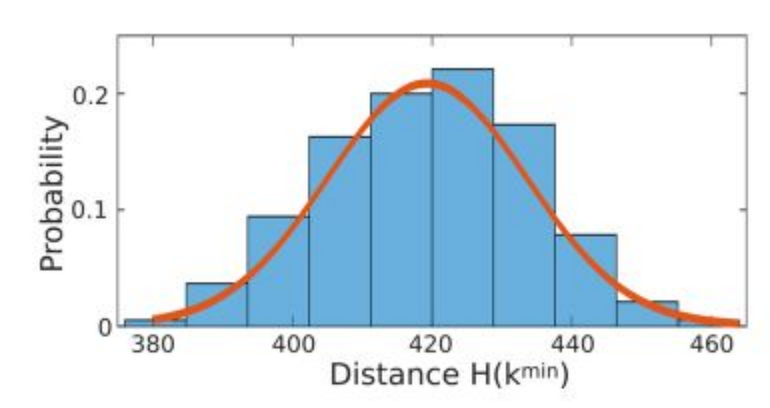
Noise under Boolean Learning

- Some experiments have been made to evaluate how noise affects learning and learning rules. To do this a network with DMD's and Boolean learning has been developed to study the effects of noise.
- The Network implemented was RC network and as usual only the readout layer was trained.
- The physical constraint ar the same as in the Reinforcement Learning Network that is positive weights and $\cos^2$ as non linear activation function.



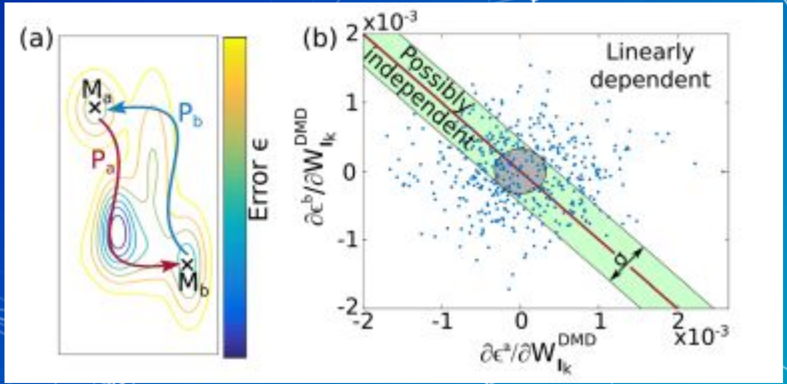
Noise under Boolean Learning

- Studying how different initializations of the same net lead to different local minima it was found that data shows a very specific and unusual error landscape topology: local minima appear not to be irregularly distributed, nor located in a particular region. Instead, the negligible correlations between the minimas' locations, and the systematic and narrow distribution of inter-minima distances reveals their almost uniform distribution across the error landscape.



Noise under Boolean Learning

- Learning does only modify readout connections and therefore neither modifies $\tilde{x}$ i nor the system's noise, making both independent of learning.
- We can therefore assume that modifications to y out induced by learning and noise remain constant for all k, hence $\sigma^l(k) = \sigma^l$ and $\sigma^n(k) = \sigma^n$.
- The Hamming distances evolution is therefore governed by noise quantified through constant $\tilde{C}$, and by how the learning algorithm picks weight $W_{l(k)}^{DMD}$ (k) from a population with a certain ρ^{op} (k), and neither error nor gradient play a role.
- The noise influence must then depend from the weights and how these are used by the learning algorithm and the bias.



6.
Wrap-up and Conclusions

Conclusion

- Photonic Networks are in an early stage but have lots of potential thanks to high parallel speed computation and low power consumption once trained.
- Different physical networks are being developed using different physical devices.
- Different training algorithm are being tested to see the best performing one.
- Physical nets have some constraints that classical ANN’s don’t.
- Noise can be an important issue to be dealt with and affects the systems in many different ways and may have different nature.
- Photonic networks can have different applications in many fields.

References

- [1] Recent advances in physical reservoir computing: A review (Gouhei Tanaka , Toshiyuki Yamane , Jean Benoit Héroux , Ryosho Nakane ,Naoki Kanazawa , Seiji Takeda, Hidetoshi Numata , Daiju Nakano , Akira Hirose)
- [2]Computing Primitive of Fully VCSEL-Based All-Optical Spiking Neural Network for Supervised Learning and Pattern Classification (Shuiying Xiang , Zhenxing Ren, Ziwei Song, Yahui Zhang, Xingxing Guo)
- [3] Reservoir Computing Neural Networks (Claudio Gallicchio)
- [4] Reinforcement Learning in a large scale photonic Recurrent Neural Network (J. Bueno, S. Maktoobi, L. Froehly, I. Fischer , M. Jacquot , L. Larger , D. Brunner)
- [5] Learning with Deep Photonic Neural Networks (Bhawani Shankar Leelar , E. S. Shivaleela and T. Srinivas)
- [6] Boolean learning under noise-perturbations in hardware neural networks (Louis Andreoli, Xavier Porte, Stéphane Chretien, Maxime Jacquot, Laurent Larger, Daniel Brunner)

THANKS!

Any questions?

You can find me at

g.antonioli3@studenti.unipi.it