POLITECNICO DI TORINO

Corso di Laurea in Nanotechnologies for ICTs

Tesi di Laurea Magistrale

Neural networks in optical domain

Relatori
prof. Emiliano Descrovi
prof. Alfredo Braunstein
prof. Luca Dall'Asta

Candidata
Enrica Racca

OTTOBRE 2019

*

Summary

In the present work we implement a deep learning framework with diffractive layers that collectively perform digit recognition. Neural networks constitute the computing system performing deep learning, which differs from any application of artificial intelligence – enabling machines to automatically learn from experience without being explicitly programmed – in that it creates the representations essential for classification organizing them into multiple levels. Our neural network is inspired by a framework recently introduced in the literature, termed as Diffractive Deep Neural Network (D^2NN). It is physically formed by multiple layers of diffractive surfaces that collaboratively perform optical diffraction when an input image is exposed to electromagnetic radiation. During the training phase implemented on a computer through deep learning's methods, trainable parameters represented by the diffractive layers' transmission coefficients are properly adjusted. Once completed the numerical stage, the design of the framework is established: the obtained transmission coefficients provide in fact information about the thickness of the diffractive layers that, if 3D-printed and settled in their well-suited laboratory setup, would give rise to a powerful device performing digit classification at the speed of light. Optical machine learning's earlier works concerned realizations of programmable devices performing typical machine learning applications and equipped with optical components, which satisfy sought-after requirements like speed and power efficiency. In this line of work, our framework represents a pioneering innovation since, once physically fabricated, it may execute the specific task for which it is trained, exploiting no power but the effectiveness of optical diffraction through passive optical layers.

Contents

Introduction

Objects classification is one of the applications which most efficaciously have been improved by deep learning ([1]), in the same way as other innumerable functions, including speech recognition ([2], [3]), translation of words from different languages ([4], [5]), object detection ([6], [7], [8], [9]), genomics ([10], [11], [12]) and drug discovery ([13], [14], [15]).
Among the various machine learning techniques, deep learning is characterized by the property of creating the representations essential for classification, organizing them into multiple – hence the name "deep" – levels, by means of brain-inspired computing systems known as artificial neural networks (ANNs). For object recognition task, at each step of training process, the neural network is typically shown an input image, labelled with the corresponding category, and produces a final output, expressing a class or the probabilities that the input belongs to each class, upon processing by intermediate "hidden" layers. Learning involves adjusting trainable parameters of the system to improve the accuracy of the ultimate result, by minimizing a measured objective function, which expresses the distance between real and desired outputs. In standard deep-learning models, many of the layers implement non-linear input-output mappings, resulting in a comprehensive framework which performs extremely convoluted functions of its inputs and thus achieves highly sophisticated tasks.

In the present work, it is implemented an all-optical "deep"[1] learning framework with diffractive layers that collectively perform recognition of hand-written digits, from 0 to 9, provided by the MNIST dataset [16]. When a laser beam of given wavelength is focused onto an input image, the diffracted electromagnetic radiation is free-space propagated from one layer to the next one and complex-modulated by the trainable phases of diffractive planes' transmission coefficients. Each small finite element of any layer implements Huygens-Fresnel principle [17], acting as source of a secondary wave when reached by luminous disturbance, and represents the computing unit of the network. By implementing the angular spectrum method, light is propagated between any two consecutive layers, from the input to the output one. This latter incorporates ten photodetectors,

[1]For reasons detailed in Section 2.1, this specific neural network is not strictly "deep", in the classical sense of the term

each univocally associated to a specific class: the system is trained to focus light onto the one which identifies the nature of the input digit. Due to implementation of linear optical functions in any layer dedicated to learning – excluded the output one which is, in fact, purely involved in detection – this structure definitely represents an attempt on a trial basis for learning, in the context of neural networks theory addressing use of non-linear modulus. The quality of performance, quantified by an accuracy specifically defined as the percentage of correctly predicted examples, constitutes a real unknown. Confidence in promising results is actually encouraged by the recent innovative achievements obtained by Lin et al.([18]): they numerically implemented and physically realized a multilayered linear framework, performing digit recognition, which constitutes, indeed, the inspiration for our work. However, their structure implements light propagation in real-space domain at $0.4\ THz$, in contrast to our Fourier transforms-based description of electromagnetic with wavelength of $532\ nm$; moreover, it slightly differs from ours in the layout and some of its operations result ambiguously defined in their paper and in the associated supplementary material [19].
Expecting performances lower than the ones of standard state-of-the-art deep learning systems ([20], [21]), which instead involve implementation of non-linear complicated functions allowing achievement of extremely intricate tasks, denotes the innovative nature of the present network, whose exploration is essentially motivated by the view of its physical realization. Once completed the training phase, the design of the framework is fixed: according to elementary electromagnetic wave's notions, the trained transmission coefficients provide information about the thickness distribution of diffractive layers. The latter could be 3D-printed and settled in their well-suited laboratory setup, giving rise to a powerful device performing digit classification at the speed of light. The fabrication technique would consist in two-photon lithography, allowing high resolutions suitable for featuring each diffractive surfaces' computing unit, expected to require relevantly small size when implementing propagation of light at such relatively short wavelength. Any parameter characterizing the neural network must be setted in view of its potential experimental implementation, which clearly requires satisfaction of specific conditions detailed by laboratory tools. Meeting demands detailed by experimental setup along with numerically pursuing the best possible performances of the machine, during its training, constitute the core of the present study, based on a fundamental harmony between optical phenomena regulating image processing, deep-learning methods for object classification and geometrical issues of structural design.

The explored framework lies in the line of work of neural networks in optical domain, a research field largely explored since 1985, when Farhat and Pissaltis published a paper [22] introducing the idea of implementing an ANN through optical interconnections, which provide considerable advantages with respect to electronic ones, including power efficiency and capability to realize multiple interconnections and simultaneous parallel calculations at the speed of light. The number of interconnections does not affect, in fact, optical signals, which, furthermore, propagate in three-dimensional free space without

limitations. Many works in the following years, including [23], [22], [24], have explored the realization of artificial neural networks based on photonics; nevertheless, nonlinear functions have usually been implemented electronically, due to difficulty in achieving them by integrating high-power lasers and optical components. Systems with continuously modifiable connections, including an optical implementation of learning networks with volume holograms storing large number of interconnections [25], have been devised as powerful physical realizations of ANN, still requiring electrical power for dynamically regulating weightings of connections.
More recently, many artificial neural networks in optical domain have been based on nanophotonics: the combination of the latter with ANN is in fact exhibiting and promising achievements in different fields, such as optical imaging, automatic optical microscopy and inverse design of photonic devices. Deep learning systems based on nanopthotonics can be realized by implementing typical operations of neural networks through optical or optoelectronic components. Computation of non-linear functions is usually achieved by photodetectors ([26], [27], [28], [29]), electro-optic modulators ([29]), light sources (LED and laser, [28]), optical amplifiers ([30]); interconnections by free space ([27]) and waveguides ([26], [29], [28]); weighting operations can be realized, instead, with holograms ([27]) and resonators ([28]).
The late ANN numerically implemented and phisically realized by Lin et al.([18]) is characterized by basing on *all* passive elements, in comparison with previous artificial neural networks composed of optoelectronic neurons, and has recently motivated strictly related studies (including [31]).
The extraordinary feature motivating this study is thus the realization of a physical device successfully performing a specific task, upon design based on training the ANN it intrinsically represents, entailing low power consumption, since integrating no active component but a mere external laser source.

The present text is structured as follows.
Chapter 1 is dedicated, in Section 1.1, to basic notions of neural networks, in the context of the more general field of machine learning, and, in Section 1.2.1, to fundamental physical principles governing electromagnetic waves' propagation.
In Chapter 2 the core of our research is presented: starting with Section 2.1 briefly introducing the working principle of the explored diffractive deep neural network, the specific features characterizing its training are gradually exposed in 2.2. This latter enhances the crucial exploration, described in Section 2.3, of physical parameters and their consequent results, reflecting the salient points of our analysis, which has lead to the reaching of layers' thickness masks, ready to be physically implemented.
Theory about the planned process of fabrication, two-photon lithography, is provided in Section 3.1 of Chapter 3, which also includes, in Section 3.2, necessary analysis on resolution issues, preceding the potential fabrication of the device.
Conclusions are provided in 3.2 and, finally, Appendix 3.2 contains generalities about diffraction together with its analysis within a specific configuration, exploited in our study.

Chapter 1

General theory

In order to lay the foundation for the explored diffractive deep neural network, some basic concepts of neural networks and electromagnetic field theory are introduced in the present chapter.

1.1 Basic concepts of neural networks

In this general introduction attention will be mostly focused on the specific deep learning's application of interest: image classification.

1.1.1 Deep learning: a subset of machine learning

Machine learning is a subset of artificial intelligence, consisting of the scientific study of algorithms and statistical models that computer systems use to perform a specific task, relying on patterns and inference. Machine learning algorithms are used in a wide variety of applications which include, besides the here considered image classification ([32], [33], [34], [35]), language translation ([36], [37], [38], [39]) transcription of speech into text ([40], [41], [42]), spam-email filtering ([43], [44], [45], [46], [47]), prognosis of possible drug molecules ([48], [49],[50]) as well as genetics and genomics ([51], [52], [53]).
Frequently, the above-mentioned applications pervading several aspects of modern society, utilize methods from a subset of machine learning, *deep learning* [1]. The basic models of general machine learning require some meticulous external guidance that converts the input data into a proper feature vector, used by the learning system to classify patterns. On the other hand, deep learning's methods enable a machine to automatically determine the representations necessary for a specific task with multiple levels of representation.
More specifically, the latter applications make use of a layered structure of algorithms named *artificial neural network* (ANN), whose design is inspired by the biological neural

network of the human brain.

1.1.2 Neural networks

The apparently simple human task of recognizing objects is performed by a brain adapted during hundreds of millions of years of evolution to perceive the real world. Visual information is processed in each brain's hemisphere by the visual cortex, composed of the primary visual cortex V1. It contains 140 million neurons with tens of billions of connections between them and receives the sensory inputs from the thalamus, together with visual areas V2, V3, V4, V5 which enhance image processing's complexity and performance.
Despite the shapes specifically characterizing an item are easily recognized by humans, they are problematic to be expressed algorithmically. Indeed, a rich variety of patterns characterizes every single object and even the apparently uncomplicated handwritten digits cannot be rigorously recognized with rules, without falling into a myriad of exceptions.
A solution to the above mentioned problem is carried out by artificial neural networks: a collection of units, termed as neurons, that pattern the operations performed by neurons in biological brain and are organized in layers linked between them with weighted connections, which transmit information through the network like synapses do in a biological brain. The set of techniques employed in neural networks allows a machine to be fed with input data and to automatically devise the representations needed for classification, as happens in every representation learning method, but with the distinguishing feature of involving multiple levels of representation, by analogy with visual cortices.
Basically, artificial neurons receive input data (in this case, images), combine linearly them with an optional threshold (bias) and generate an output through an activation function. The desired task, such as image classification, is carried out by the terminal outputs. Neurons of one layer connect only to neurons of the two adjacent layers, namely the previous one and following one. The so called *input layer* receives external data, the *output layer* generates the final result and all the layers in between these two are termed as *hidden layers*. Each connection transfers the output of one neuron as an input to another neuron and, carrying a weight, allows the computation of the input to a neuron as a weighted sum of the previous neurons' outputs.
Networks allowing connections between neurons of the same layer are called *recurrent networks*, while in *feedforward network* the connections form a directed acyclic graph. The activation function is devised so that a small change in input produces a small change in output and is non-linear.
To highlight the relationship with biological neurons, it should be noted that in these the action potential, occurring when a neuron sends information down an axon, fires only if the potential difference at membranes overcomes a given threshold. An analog principle holds for artificial neurons, but in this case the response to input data depends on the specific used activation function and is adapted to the precise role the neuron and the network are intended to accomplish.

It is emphasized that, for complex tasks of pattern recognition typically performed through deep learning, the activation function is required to introduce a non-linear factor in the network. The reason for this can be understood by considering the effect of a linear classifier, currently employed in part of machine learning applications in concomitance with ad-hoc devised features.
A two-class linear classifier computes a weighted sum of its input components: if the weighted sum is above a given threshold, the input is categorized into a specific class. Such operation can be imagined as the separation of a multidimensional input space into half-spaces by an hyperplane, as sketched in Fig.1.1: all the points in the region on a side of the hyperplane are classified as "belonging to that category (e.g., red rhombus)", the others as "not belonging to that category (e.g., blue star)".

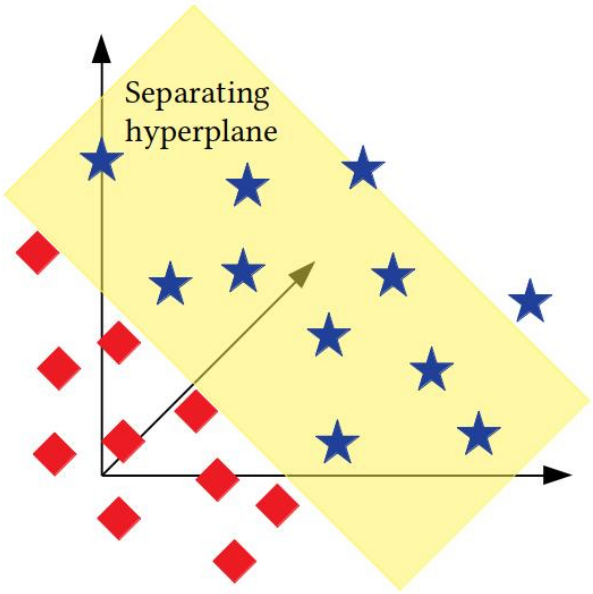


Figure 1.1: Two-class (red rhombuses and blue stars) linear classifier

As mentioned above, "shallow" classifiers (like the linear one) are not sufficient on their own in any machine learning procedure. This is due to the fact that any application involving recognition requires the output function to be both insensitive to small variations of input data and sensitive to specific details which may determine the correct classification of an element. More specifically, image classification needs the output function to be insensitive to small variations in orientation and position of the input object and simultaneously sensitive to specific tiny variations, in order to distinguish, as an example, two different items in the same position and surrounded by a similar background. By means of multiple non-linear layers, it is possible to extraordinarily implement convoluted functions maximizing both selectivity and invariance. The multilayered structure,

proper to deep learning, precisely involves modules computing non-linear functions of their input and is designed to automatically learn by means of adjustable parameters, whose operation will be detailed later. This last key characteristics allows to avoid hand-designed feature extractors and their associated refined expertise, which in non-automatic learning need instead to be combined with linear classifiers.
From Fig.1.2, it is possible to observe how a neural network, in this simple case including only two input units, two hidden units and a hidden unit, can make the classes of data linearly divisible: the input space exhibiting a regular grid with categories of data on the blue and red lines (left) is altered by the multilayered framework to linearly separate the two classes (right).

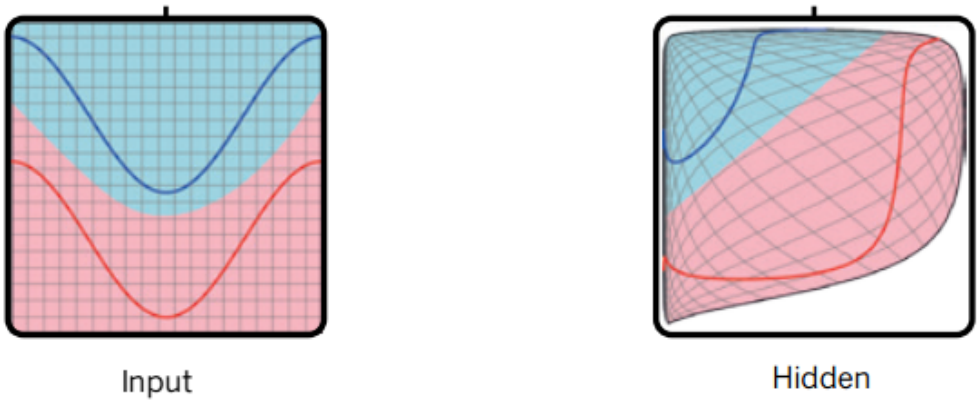


Figure 1.2: Input space's distortion in a generic multilayer neural network; figure taken from [1]

Neural networks performing image classification are typically based on supervised learning: the input data the machine is fed with during training, termed as *training set*, is in this case composed of images and their corresponding label. The same holds for the *test set*, a different set of examples used to compute the performance of the machine after training; it contains new images with respect to the training set, with the purpose of testing the outputs of the system to inputs never seen during training.
The output layer has typically as many neurons as the number of provided classes: considering classification of digits from 0 to 9, it will therefore have 10 elements, each of them associated to a different digit and representing the value of probability for the input image to belong to that specific number. Stated differently, the output vector assigns each category a *score* that quantifies how much the considered image is likely to belong to that specific class.
As regards the trainable parameters that during training regulate themselves until correct classification is performed with the desired accuracy, they consist of the weights of connections (denoted as w in the following) and optionally of the biases (b) added to the summed inputs to a neuron contributing to the argument of the activation function. Small changes in such parameters must produce small variations in the output of the network,

to make the training properly work. In fact, assuming that the system is erroneously classifying an "8" as a "3", it is intuitive that changes in weights and optionally biases need to be small, so that the network gradually learns to classify the image as an "8" and the behavior of the rest of the whole structure is not compromised. Instead, large variations of trainable parameters as well as binary neurons producing "0" or "1" as an output could modify the rest of the structure in some complicated way. It is thus required to use a smooth function, ensuring that a small change Δw_j in the weight linking neuron j to neuron i and Δb_i in the bias of unit i produce a limited variation $\Delta output_i$ in that neuron's output, which can thus be properly approximated by:

$$\Delta output_i \approx \sum_j \frac{\partial output_i}{\partial \Delta w_{ji}} \Delta w_{ji} + \frac{\partial output_i}{\partial \Delta b_i} \Delta b_i \tag{1.1}$$

where the sum is extended to all the neurons j connected to the unit i. The linearity of the the function "$output_i$" with respect to Δw_j and Δb_i easily allows to obtain any requested tiny variation in the output through small changes in the trainable parameters.
In this way, a good classification of input data can be obtained; its achievement is quantified by the so called *objective function* or *cost function*, calculating the error (i.e. a difference) between the obtained output and the target output. The latter could, for instance, be a canonic array, with a 1 in correspondence of the element referring to the correct class and remaining components equal to 0.
The aim of the algorithm becomes thus to change weights and biases in order to minimize the objective function. This is commonly optimized with the stochastic gradient descent (SGD) iterative method, which consists of providing the machine an input set of examples, computing the errors deriving from the outputs, evaluating the average (over a given set of examples) negative gradient, which indicates the direction leading the objective function closer to a minimum, and correspondingly arranging the weights. It is considered as a stochastic approximation of a gradient descent method, because, at each step of the iterative process, it is calculated the average gradient over a small set of examples, which is an estimation of the actual average gradient over the entire training set. At each step it is taken a new small set of data and the process is repeated until the average of the objective function stops decreasing.
Considering, for the sake of simplicity, only the set of weights $\vec{w}$ as trainable parameters, the objective function E to be minimized is expressed as

$$E(\vec{w}) = \frac{1}{n} \sum_{i=1}^{n} E_i(\vec{w})$$

where n corresponds to the cardinality of the whole training set and E_i is the error associated to an input. The cost function consists generally of a mean squared error:

$$E(\vec{w}) = \frac{1}{2n} \sum_{i=1}^{n} |output(i) - t(i)|^2 \tag{1.2}$$

with $output(i)$ and $t(i)$ denoting respectively the vectors of true outputs and target outputs obtained with the i^{th} input image.
The SGD method performs the following change of weight w_{jk} for the connection between the j^{th} and the k^{th} neuron of two consecutive layers:

$$\Delta w_{jk} = -\eta \sum_{i=1}^{m<n} \frac{1}{m} \frac{\partial E_i}{\partial w_{jk}} \tag{1.3}$$

where m is the cardinality of a small set of data and $\eta > 0$ is a step size, called *learning rate*.
It is now presented with greater detail the working principle of a specific multi-layer network, namely the feedforward network represented in Fig.1.3, with one hidden layer containing three neurons and the input and output layer with respectively five and two units. The considered structure is defined as *fully connected*, meaning that every neuron of one layer linked to every neuron of the next layer; however, different patterns of connection are in principle be allowed in a neural network.

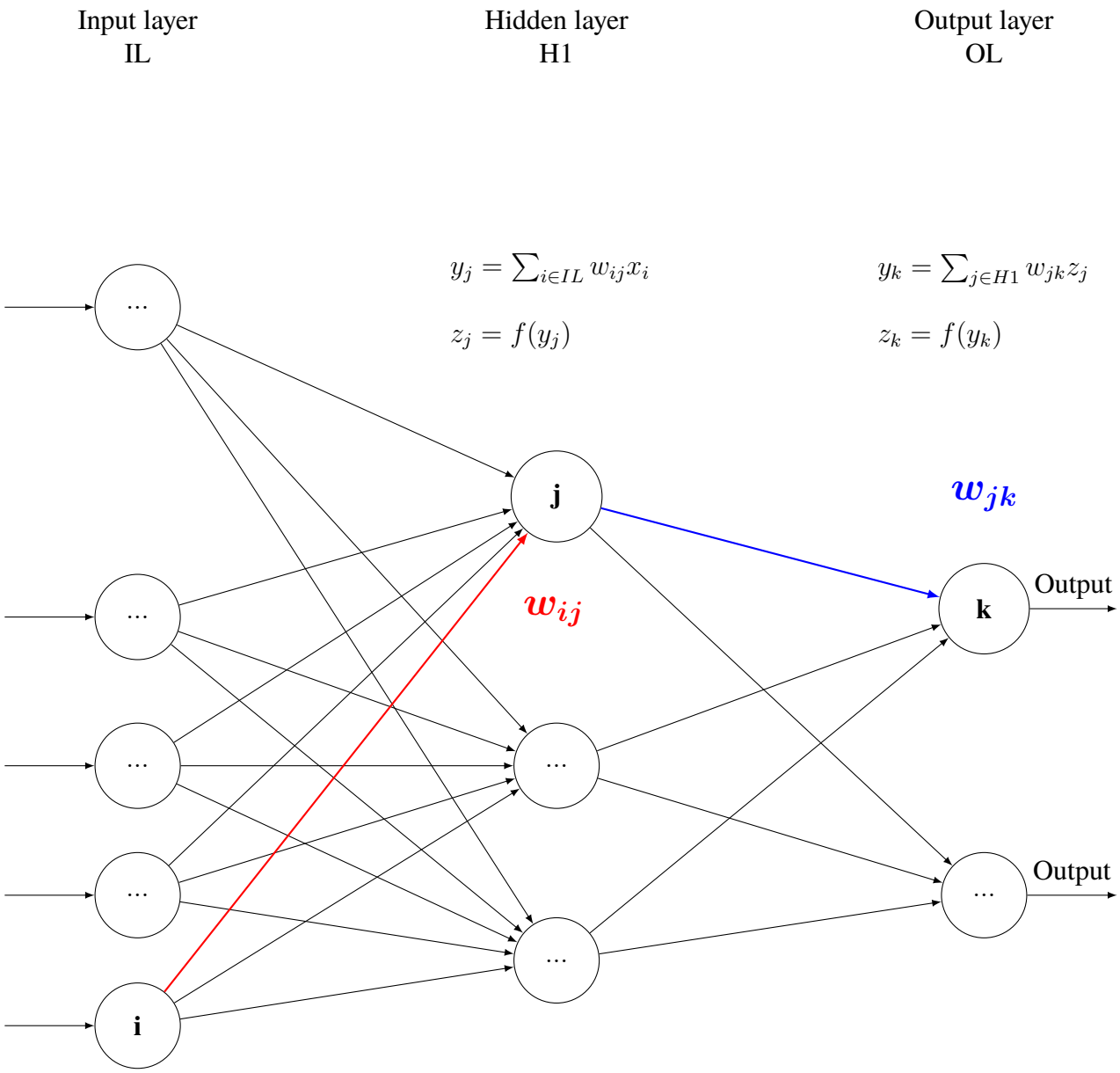


Figure 1.3: Forward pass in a multilayer neural network

The j^{th} neuron, representing a computation unit of the hidden layer H1, performs the following operations:

- it combines the input data coming from the previous layer with specific weights, so

that the total input y_j to the j^{th} neuron is a weighted sum of the outputs from nodes of the previous layer, namely:

$$y_j = \sum_{i \in IL} w_{ij} x_i$$

where w_{ij} is the connection directed from the i^{th} neuron (of the input layer IL) producing the output x_i, to the j^{th} node of H1. The sum is extended to all the neurons of IL since the network is fully connected; in a general case, it would be extended to $M \leq N$ neurons of the input layer;

- it applies a nonlinear function $f(.)$ to its total input y_j:

$$z_j = f(y_j)$$

[1] where the non-linear function f may be, according to the most common choices of the literature, the rectified linear unit $f(y) = max(0, y)$, the sigmoids such as the hyberbolic tangent $f(y) = (e^y - e^{-y})/(e^y + e^{-y})$ and the logistic function $f(y) = 1/(1 + e^{-y})$.

The gradients requested for the minimization of the objective function in the SGD method, or generally in gradient-based optimization algorithms, are computed through the *back-propagation method*, if the modules implemented by each layer are smooth enough functions; this explains the previously presented typical choices of activation function. The backpropagation procedure is merely an application of the chain rule for derivatives and it allows to obtain the derivative of the objective function E with respect to each weight (with reference to Eq.1.3, $\frac{\partial E_i}{\partial w_{jk}}$ for weight w_{jk}) by knowing the gradient of E with respect to the input of a module, which is in turn computed by "backpropagating" the gradient with respect to the output of that module. Basically, by starting from the output layer and progressively getting to the input layer, the derivatives are evaluated at each module.
The method is described in Fig.1.4, which refers to the same neural network of Fig.1.3.

[1] if the trainable parameter given by the bias b_j of neuron j were, differently from here, considered, it would add up to the input y_j in the argument of $f(.)$, giving as a result $z_j = f(y_j + b)$

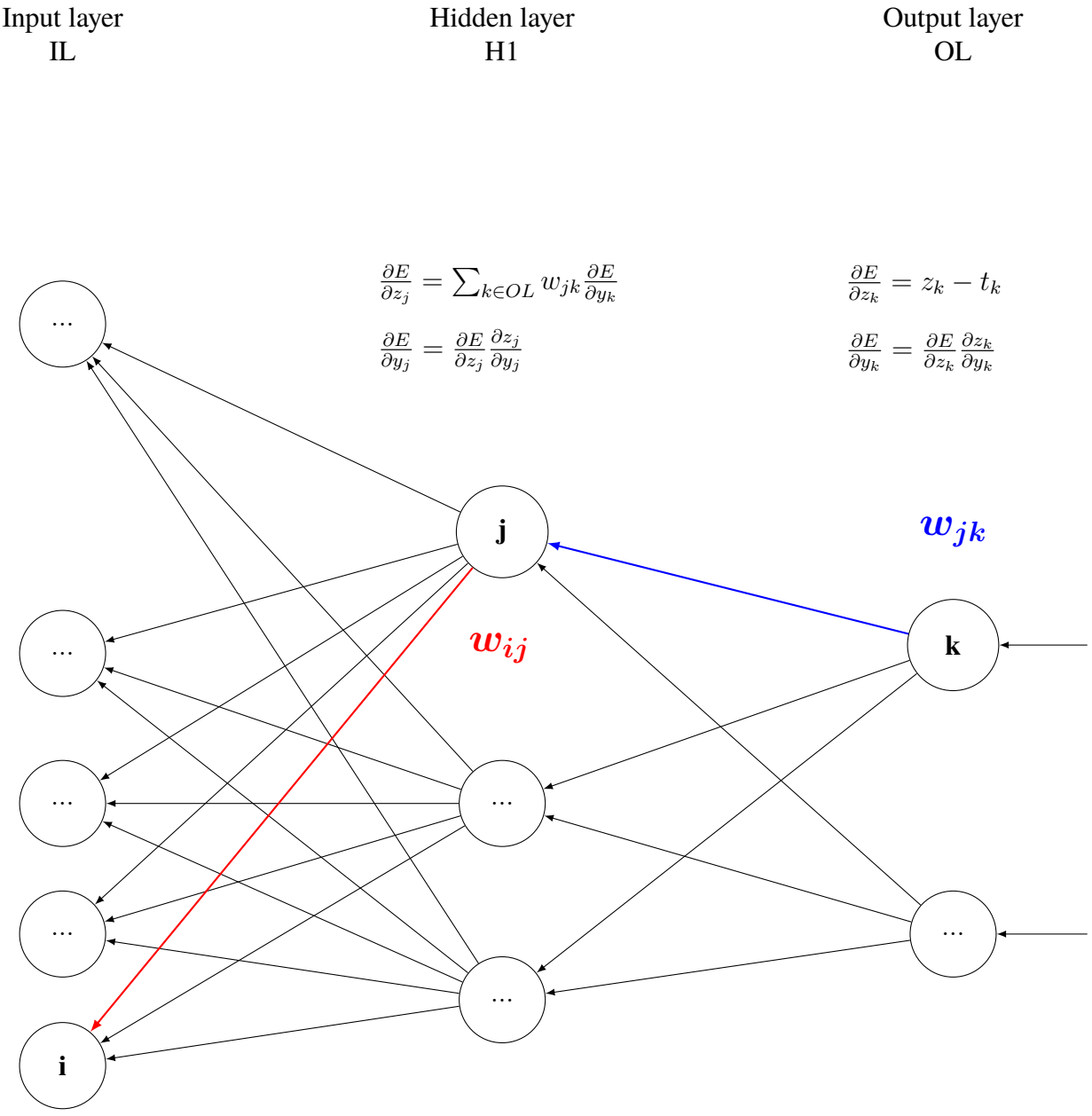


Figure 1.4: Backpropagation in a multilayer neural network

In correspondence of the neuron k at the output layer the derivative of the cost function E with respect to the output is given by $\frac{\partial E}{\partial z_k} = z_k - t_k$, considering $E = \frac{1}{2}(z_k - t_k)^2$, where t_k is the target output of neuron k. The derivative of E with respect to the input y_k

of that neuron will be instead given, according to the chain rule, by: $\frac{\partial E}{\partial y_k} = \frac{\partial E}{\partial z_k}\frac{\partial z_k}{\partial y_k}$, where $\frac{\partial z_k}{\partial y_k} = \frac{\partial f(y_k)}{\partial y_k}$. Once $\frac{\partial E}{\partial y_k}$ is known, the derivative of the cost function with respect to the weight w_{jk} is simply given by $\frac{\partial E}{\partial w_{jk}} = z_j \frac{\partial E}{\partial y_k}$, since $y_k = \sum_{j \in H1} w_{jk} z_j$. The gradients are propagated backwards at each module until the input layer is reached, as shown in the considered figure.

1.2 Underlying concepts of electromagnetic field theory

This section is dedicated to theoretical tools essential to model electromagnetic propagation in the neural network of interest.

1.2.1 Fourier optics

The study of classical optics using Fourier transforms goes under the name of Fourier optics. It is well-known that both the electric field $\vec{E}$ and the magnetic field $\vec{H}$ obey the wave equation, obtained by manipulating the Maxwell's equations. Considering the electric field $\vec{E}$, the wave equation is:

$$\nabla^2 \vec{E}(\vec{r}, t) = \frac{n^2}{c^2} \frac{\partial^2 \vec{E}(\vec{r}, t)}{\partial^2 t} \tag{1.4}$$

where n and c are respectively the refractive index of the medium and the speed of light in vacuum; $\vec{r}$ and t denote the spatial position and time.
In particular,

$$n = \frac{\epsilon}{\epsilon_0}$$

and

$$c = \frac{1}{\sqrt{\epsilon_0 \mu_0}}$$

with ϵ, ϵ_0 and μ_0 referring respectively to the absolute permittivity of the medium, the vacuum permittivity and the vacuum permeability.
An identical scalar wave equation is obeyed by the components of the vectors $\vec{E}$ and $\vec{H}$, therefore a generic electric field's component E_i obeys the equation:

$$\nabla^2 E_i(\vec{r}, t) = \frac{n^2}{c^2} \frac{\partial^2 E_i(\vec{r}, t)}{\partial^2 t} \tag{1.5}$$

The same holds for any component H_i of the magnetic field, therefore the behavior of all the components of $\vec{E}$ and $\vec{H}$ can be written in a compact form through a single scalar wave equation

$$\nabla^2 u(\vec{r}, t) = \frac{n^2}{c^2} \frac{\partial^2 u(\vec{r}, t)}{\partial^2 t} \tag{1.6}$$

For a monochromatic wave, the scalar field $u(\vec{r},t)$ solution of Eq.1.6 is

$$u(\vec{r},t) = \Re\{U(\vec{r})F(t)\} \tag{1.7}$$

where $\Re\{\}$ denotes the "real part", $U(\vec{r})$ is the complex function of spatial position $\vec{r}$ and $F(t)e^{-j2\nu t} =$ is the complex function of time, with ν denoting the frequency of the wave. The factorization emerging in Eq.1.7 separates the spatial and the temporal part of the scalar field $u(\vec{r},t)$ and allows to reduce complicated two-dimensional mathematical manipulations to simpler one-dimensional manipulations. Moreover, since the time dependence is generally known a priori, the analysis can be focused on the spatial propagation for a complete description of the field.

The general solutions of Eq.1.6 do not provide practical and effective information about a specific case of interest, implemented in the neural network later presented in this document, which is the evolution of light between two different planes, parallel one with the other and both perpendicular to the direction of propagation, termed as *optical axes*. To examine this, it is necessary to follow the mathematical analysis presented below.
It is considered the Cartesian coordinate system of Fig.1.5, with optical axes oriented along the positive z direction and the wave incident on a transversal plane (x,y). The complex field across that plane, positioned at $z = 0$, is represented by the complex field $U(x,y,0)$ and the purpose is to calculate the field distribution that appears in correspondence of a second plane parallel plane at a distance z to the right of the first plane.

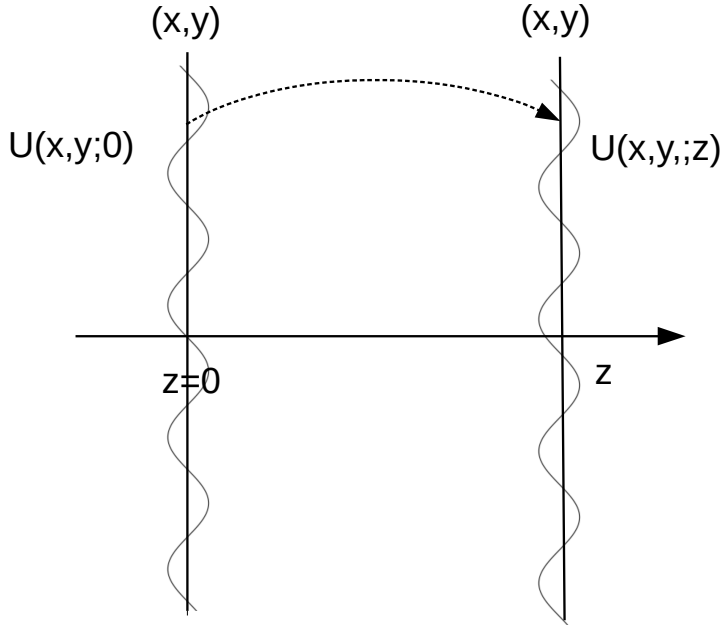


Figure 1.5: Field distributions $U(x, y; 0)$ in correspondence of plane (x, y) at $z = 0$ and $U(x, y; z)$ in correspondence of a second plane (x, y) at distance z

To this purpose, it is firstly considered the two-dimensional Fourier transform of $U(x, y,0)$, $\mathcal{F}\{U(x, y,0)\}$:

$$A(f_x, f_y; 0) = \iint_{-\infty}^{+\infty} U(x, y,0)\ e^{-j2\pi(f_x x + f_y y)}\ dx\, dy = \mathcal{F}\{U(x, y,0)\} \tag{1.8}$$

The complex function $A(f_x, f_y; 0)$ of spatial frequencies f_x, f_y is named *angular spectrum* and expresses amplitude of each planewave in which the field $U(x, y,0)$ is decomposed. To highlight the physical meaning of the angular spectrum, it is useful to consider its inverse Fourier transform:

$$U(x, y,0) = \iint_{-\infty}^{+\infty} A(f_x, f_y; 0)\ e^{j2\pi(f_x x + f_y y)}\ df_x\, df_y = \mathcal{F}^{-1}\{A(f_x, f_y; 0)\} \tag{1.9}$$

The integral above is clearly extended to a set of planewaves, each with an amplitude corresponding to a couple of values of spatial frequencies (f_x, f_y).
It is now considered the the spatial part of a simple planewave $w(\vec{r}, t)$:

$$w(\vec{r}) = e^{i\vec{k}\cdot\vec{r}} \tag{1.10}$$

where $\vec{k} = \frac{2\pi}{\lambda}(\alpha, \beta, \gamma)$ is the wave vector with magnitude $|\vec{k}| = \frac{2\pi}{\lambda}$ and direction cosines (α, β, γ) with $\gamma = \sqrt{1 - \alpha^2 - \beta^2}$, represented in Fig.1.6:

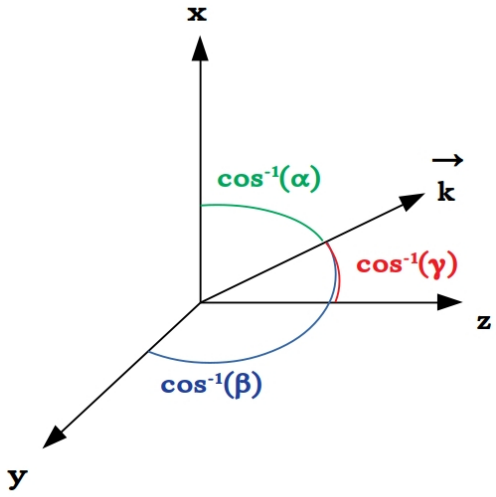


Figure 1.6: Wave vector $\vec{k} = \frac{2\pi}{\lambda}(\alpha, \beta, \gamma)$ with magnitude $|\vec{k}| = \frac{2\pi}{\lambda}$ and direction cosenes (α, β, γ)

It is clear the complex function $A(f_x, f_y; 0)\ e^{j2\pi(f_x x + f_y y)}$, in correspondence of the plane $z = 0$, can be viewed as a planewave with amplitude $A(f_x, f_y; 0)$ and direction cosines related to spatial frequencies:

$$\left(\alpha = \lambda f_x, \beta = \lambda f_y, \gamma = \sqrt{1 - \lambda f_x^2 - \lambda f_y^2}\right) \tag{1.11}$$

As previously emerged and as highlighted in 1.11, the wave vector $\vec{k}$ may be also written as a function of spatial frequencies f_x, f_y, f_z:

$$\vec{k} = 2\pi(f_x, f_y, f_z) \tag{1.12}$$

The angular spectrum of the complex field U across the plane (x', y') parallel to the plane (x, y) and at a distance $z > 0$ from it is given by:

$$A(f_x, f_y; z) = \iint_{-\infty}^{+\infty} U(x, y, z)\ e^{-j2\pi(f_x x + f_y y)}\ dx\, dy = \mathcal{F}\{U(x, y, z)\} \tag{1.13}$$

In order to find out the relation between $A(f_x, f_y; 0)$ and $A(f_x, f_y; z)$ or equivalently the propagation of the angular spectrum, $U(x, y; z)$ is written as:

$$U(x, y; z) = \iint_{-\infty}^{+\infty} A(f_x, f_y; z)\ e^{j2\pi(f_x x + f_y y)}\ df_x\, df_y = \mathcal{F}^{-1}\{A(f_x, f_y; z)\} \tag{1.14}$$

and is inserted into the following Helmoltz equation [2]

$$\nabla^2 U + k^2 U = 0 \tag{1.15}$$

It is easily obtained the following differential equation for the angular spectrum $A(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z)$:

$$\frac{\partial^2 A(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z)}{\partial z^2} + \left(\frac{2\pi}{\lambda}\right)^2 (1 - \alpha^2 - \beta^2)\, A\left(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z\right) = 0 \tag{1.16}$$

whose solution is

$$A\left(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z\right) = A_0\, e^{j\frac{2\pi}{\lambda}\sqrt{1-\alpha^2-\beta^2}} \tag{1.17}$$

where $A_0 = A(f_x, f_y, 0)$. It is clear that the angular spectrum simply propagates as a planewave. A proper, in the physical sense, expression for $A(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z)$ is actually given by:

$$A\left(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z\right) = A_0\, e^{j\frac{2\pi}{\lambda}\sqrt{1-\alpha^2-\beta^2}}\, circ(\alpha^2 + \beta^2 \leq 1) \tag{1.18}$$

where the circular function

$$circ(\alpha^2 + \beta^2 \leq 1) = \begin{cases} 0, & \text{if } \alpha^2 + \beta^2 > 1 \\ 1, & \text{if } \alpha^2 + \beta^2 \leq 1 \end{cases} \tag{1.19}$$

ensures that in the expression of $A(\frac{\alpha}{\lambda}, \frac{\beta}{\lambda}, z)$ no exponential functions with negative real exponent can play a role, since they would make the angular spectrum exponentially decay, basically inhibiting propagation of light. By writing down the following more compact expression for the propagation of the angular spectrum

$$A = A_0\, e^{ik_z z} \tag{1.20}$$

,where $k_z = \frac{2\pi}{\lambda}\sqrt{1 - \alpha^2 - \beta^2}$, it is clear that the propagation for the complex field U may be devised by antitransforming both the left-hand side and right-hand side of Eq.1.21, resulting in the convolution

$$U = U_0 * h \tag{1.21}$$

where

$$U = \mathcal{F}^{-1}\{A\}$$
$$U_0 = \mathcal{F}^{-1}\{A_0\}$$
$$h = \mathcal{F}^{-1}\{e^{ik_z z}\}$$

[2] Such equation is simply obtained by substituting the expression $u(\vec{r}, t) = \Re\{U(\vec{r})F(t)\}$ into Eq.1.6 and simplifying out the temporal function $F(t)$ acting like a multiplicative factor at each member

Since $h(x,y,z) = -\frac{j}{\lambda}\frac{e^{jkr}}{r}\frac{z}{r}$, with $r = \sqrt{x^2+y^2+z^2}$, it is obtained:

$$\begin{aligned} U(x,y;z) &= \iint_{-\infty}^{+\infty} U_0(x',y';0)\ h(x-x',y-y',z)\ dx'\,dy' = \\ &= -\frac{j}{\lambda}\iint_{-\infty}^{+\infty} U_0(x',y';0)\ \frac{e^{jkr}}{r}\ \frac{z}{r}\ dx'\,dy' \end{aligned} \tag{1.22}$$

with $r = \sqrt{(x-x')^2+(y-y')^2+z^2}$.
Eq.1.22 is the mathematical expression of the Huygens-Fresnel principle and expresses the field $U(x,y;z)$ as a superposition of diverging spherical waves $\frac{e^{jkr}}{r}$ from secondary sources located at each point of coordinates $(x',y';0)$ on the (x',y') plane.

In summary, the propagation of an electromagnetic wave perpendicularly incident on two parallel planes, one positioned at $z = 0$ and the second at a distance $z > 0$ from it, can be performed in one of the two following equivalent ways:

- working in the frequency domain, that is:
  - performing the Fourier transform
  $$A(f_x,f_y;0) = F\{U(x,y,0)\}$$
  - propagating the angular spectrum simply by performing a phase shift of entity $e^{jk_z z}$, namely
  $$A(f_x,f_y;z) = A(f_x,f_y;0)\,e^{jk_z z}$$
  - finally finding the electromagnetic field at (x,y,z) by antitransforming the propagated angular spectrum
  $$U(x,y,z) = F^{-1}\{A(f_x,f_y;z)\}$$
- working in the real domain by implementing the Huygens-Fresnel principle:

$$\begin{aligned} U(x,y;z) &= \iint_{-\infty}^{+\infty} U_0(x',y';0)\ h(x-x',y-y',z)\ dx'\,dy' = \\ &= \iint_{-\infty}^{+\infty} U_0(x',y';0)\ \frac{e^{jk_z z}}{r}\ \frac{z}{r}\ dx'\,dy' \end{aligned} \tag{1.23}$$

In the diffractive neural network presented in the next chapter, the first approach will be followed.

1.2.2 Relating optical intensity and complex scalar field $U(\vec{r})$

Since the directly measurable quantity in optics is optical power (proportional to optical intensity, which is the optical power per unit area), it is essential to associate it to the complex scalar field $U(\vec{r})$ introduced in the paragraph 1.2.1, as well as to the the electric field $\vec{E}$ and magnetic field $\vec{H}$.
The key linking element is the Poynting vector $\vec{S}$, corresponding to the directional energy flux, namely the energy transfer per unit area per unit time, of an electromagnetic field. $\vec{S}$ is defined as the vector product of $\vec{E}$ and $\vec{H}$:

$$\vec{S} = \vec{E} \wedge \vec{H} \tag{1.24}$$

The SI unit of Poynting vector is the watt per square meter, namely $\frac{W}{m^2}$.
Therefore, its time average represents the intensity of the electromagnetic field, namely:

$$\left\langle \vec{S} \right\rangle = I \tag{1.25}$$

with SI unit given by watt per square meter ($\frac{W}{m^2}$).
Since the modulus of the magnetic field $|\vec{H}|$ and of the electric field $|\vec{E}|$ are related as follows:

$$|\vec{H}| = \sqrt{\frac{\epsilon_r \epsilon_0}{\mu_r \mu_0}}\, |\vec{E}| = \sqrt{\frac{\epsilon}{\mu}}\, |\vec{E}| \tag{1.26}$$

with ϵ_r, μ_r, ϵ_0, μ_0 denoting respectively the relative dielectric permittivity and the relative permeability of the medium, the vacuum permittivity and the vacuum and permeability, it holds that

$$|\vec{S}| = \frac{\epsilon_r \epsilon_0}{\mu_r \mu_0}\, |\vec{E}|^2 = \frac{\epsilon}{\mu}\, |\vec{E}|^2 \tag{1.27}$$

The proportionality of the Poynting vector, whose time average represents an intensity, to the squared modulus of the vector $\vec{E}$ leads to define the intensity of a scalar field at position $\vec{r}$ as the squared modulus of the complex field $U(\vec{r})$:

$$I(\vec{r}) = |U(\vec{r})|^2 \tag{1.28}$$

In the present work, the complex scalar field U will identify the electromagnetic wave propagating between any two layers. Analysis concerning the distribution of light onto any plane will be, instead, performed through its directly measurable intensity I.

Chapter 2

Diffractive "deep" neural network

Defined the theoretical basis of our study, it is now possible to get to the core of it. The aim is to implement a deep learning framework with diffractive layers that collectively perform digit recognition.
It is worthy to emphasize from the beginning that any aspect, later described, of the network has been devised in view of its experimental implementation.

2.1 Introduction to the study

As previously mentioned, the studied neural network is inspired by the framework defined as "D^2NN" (Diffractive Deep Neural Network), recently introduced by Xing Lin et al. in [18] and detailed in the associated supplementary material ([19]). The aim is to train a neural network that can perform digit classification through deep learning's method, with the innovative feature of being all-optical.
Some unique differences in its architecture with respect to standard neural networks will be emphasized in the course of the description. Nevertheless, it is necessary to immediately point out a singular aspect: the network is entirely modelled by physical electromagnetic propagation, therefore no neuron applies non-linear function to its input data. The layers from the input to the last hidden one (included) are simply passive elements providing an obstacle to light propagation, thus, by definition, "diffractive" elements causing the angular spectrum's broadening of the electromagnetic wave incident on them. A separate discussion is instead required for the output layer, in the following also named as "detector plane", which is not considered as part of the "diffractive layers".

Actually, referring to the multilayer diffractive setup as a *deep* neural network is misleading and improper, as reported in the comment [54] on [18]. In fact, since each layer performs purely linear optical functions, the entire optical system can be described by a

single-layer structure (hence a more proper use of quotes around "deep"), far from a real neural network, like the ones exposed in 1.1.2, which is classically referred as *deep* for necessarily relying on a multilayered framework composing non-linear functions, with the purpose of executing extremely complicated tasks. It is in fact clear that composition of more linear functions corresponds to a unique linear one and this is reflected, in the neural network, in the abstract collapse of the multilevel structure into a single layer.
Nevertheless, piling up more layers implementing linear functions makes performances of the structure improve. As will be detailed shortly, in our framework any diffractive layer contains precisely N^2 trainable parameters, each of them associated to a neuron and multiplied to the output coming from that specific neuron (for the sake of clarity, it is again highlighted that, on the other hand, in classical deep learning models the trainable parameters are essentially the weights regulating the inputs to each neuron of any hidden layer; as a result, this latter, in a fully-connected network, will contain $N^2 \times N^2$ trainable parameters). Incorporating multiple planes in our specific framework proves to increase classification accuracy, precisely for the involvement of more trainable coefficients, specifically adjusting themselves in order to accomplish the desired task.
It is worthy to point out that, as will be clearer later, in our framework a further operation is implemented, specifically in the output layer, which acts similarly to a single-layer perceptron. A mask, represented by a matrix of zeros and ones, is indeed applied to such plane, resulting in cutting values of intensity outside the detecting regions: this allows the "observer" more easily discriminate which detector is the most luminous one. The perceptron is basically an artificial neuron using the Heaviside step function as activation function. It constitutes the basic mathematical model for ANNs, developed in the 1950s and 1960s; nonetheless, as pointed out in 1.1.2, today it is more common to use other models of artificial neurons (like the sigmoid neuron), implementing non-linear functions.
It is further emphasized that an effective all-optical *deep* neural network could be implemented by incorporating optical non-linear functions in correspondence of the various diffractive layers. Being the present multilayered framework the possible base of a potential non-linear optical system, we feel sort of entitled to refer, in the present document, to it as "deep", sometimes inappropriately neglecting quotes.

Coming back to describe the operation of the considered network, its trainable parameters are contained in the transmission coefficients of the diffractive layers and are represented by a phase value. The transmission coefficient t_i^l of the i^{th} neuron placed in layer l at spatial position (x_i, y_i, z_i) is given by:

$$t_i^l(x_i, y_i, z_i) = e^{j\phi_i^l(x_i, y_i, z_i)} \tag{2.1}$$

where the phase $\phi_i^l(x_i, y_i, z_i)$ represents the only adjustable parameter related to the considered neuron; the total number of trainable parameters in the network corresponds to the number of neurons in the whole framework, differently from a fully-connected standard neural network that approximately has as many modifiable parameters as the

square of number of neurons. The transmission coefficient $t_i^l(x_i, y_i, z_i)$ of a neuron is actually composed of both amplitude a_i^l and phase ϕ_i^l terms, namely:

$$t_i^l(x_i, y_i, z_i) = a_i^l \, e^{j\phi_i^l(x_i, y_i, z_i)}$$

therefore also the amplitude a_i^l may in principle represent a trainable parameter. Nevertheless, optical losses are neglected, therefore the amplitude is assumed to be a constant ideally equal to 1.
It is worthy to highlight a further differences with respect to standard deep neural networks, still deriving from the fact that the multiple layers are connected by means of electromagnetic radiation (mathematically described by a complex field, which in Section 1.2.1 has been denoted as U) propagating through them: the inputs to each neuron are complex-valued rather than real-valued.

During the training phase, implemented on a computer using deep learning's methods, the trainable parameters are properly adjusted; once the network has learned with the desired degree of accuracy, the numerical phase is complete and the design of the network is established by the obtained phase values. In fact, these give rise to a height map of the diffractive layers, which, once 3D-printed and settled in a laboratory setup matching all the conditions imposed before training, compose a powerful device performing the specific task it was devised for, at the speed of light and without power consumption. The height map is obtained by evaluating the thickness h of each relatively small "cell" – corresponding to a neuron – which the layer is subdivided into. With reference to basic notions of electromagnetic waves, h is given by:

$$h = \frac{\lambda\phi}{2\pi\Delta n} \tag{2.2}$$

where λ is the wavelength of the electromagnetic radiation shed onto the image to be recognized and propagating through the network, while Δn represents the difference between refractive index of the medium constituting the fabricated layers and the medium for propagation of light in the laboratory (air).

2.2 Training of the deep diffractive neural network

In this section, the training process allowing adjustment of the phase value ϕ of each neuron is gradually presented.

2.2.1 Architecture and geometry of the framework

It is firstly described the global architecture of the diffractive neural network together with its geometrical parameters. At the moment, the values of the latter are not made explicit: they will be presented in concomitance with their corresponding results, to show the

flow of work involving their successive modifications and leading to define geometrical features suitable to experimental implementation of the network.

The neural network design is implemented on Python, using Tensorflow math library. Since the network is trained as a classifier of handwritten digits from 0 to 9, supervised learning takes place. The machine is fed with "examples", each including an handwritten digit image and its corresponding label, provided by the MNIST dataset ([16]), which has a training set and a test set of respectively 60000 and 1000 examples.
Albeit the number of layers is defined while exploring the system and thus will be specified in the following, it is instead established a priori that, as previously mentioned, the input and hidden layers involve neurons with modifiable phase coefficients and form the subsystem dedicated to learn, while the output layer – in the following, also termed as "detector plane" – contains ten separated and distinguishable detector regions, basically performing the operation of classification. The neural network is in fact trained to make the light concentrate in one detector, out of ten, which is univocally associated to one specific digit. The detailed operations performed by each component of the network will be discussed the next paragraph.
Referring to a three-dimensional Cartesian coordinate system, the architecture of the network consists of a given number of squared layers, laying on (x,y) planes; they are arranged parallel and at a distance d one with respect to the other, along the z direction. As an example, an arbitrary simple structure with 3 layers is represented in Fig.2.1.

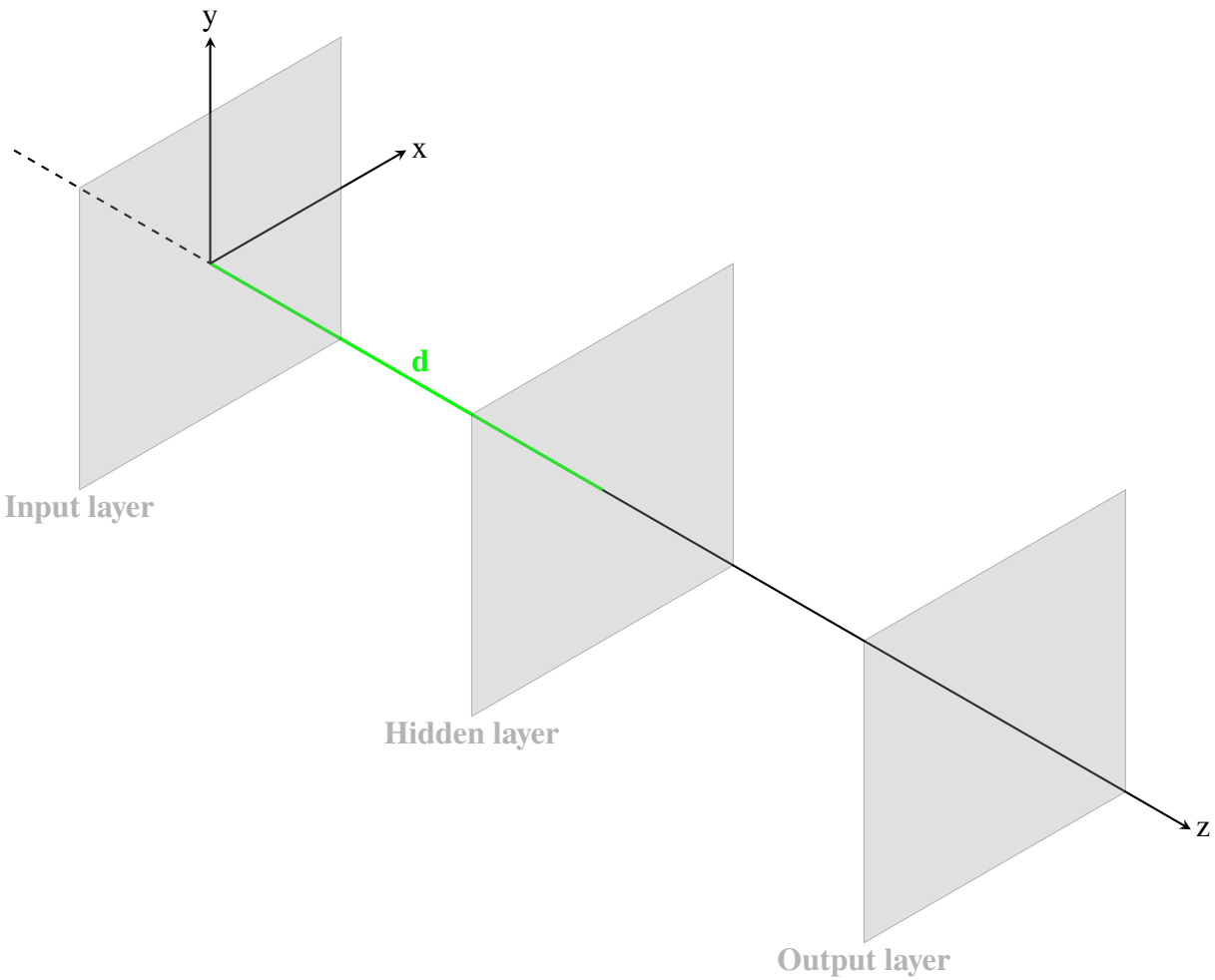


Figure 2.1: Neural network consisting of 3 squared layers, laying on (x, y) planes arranged parallel and at a distance d one with respect to the other, along the z axes

Each diffractive layer has side L and is formed by $N \times N$ neurons, identified by squared pixels with a finite size δx, as shown in Fig.2.2 for an arbitrary value of N.

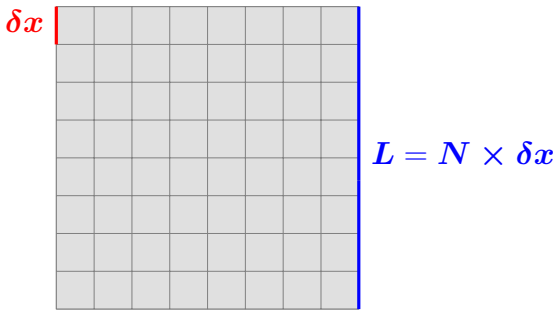


Figure 2.2: A single diffractive layer of the neural network, with side L and containing $N^2 = 8^2$ pixels of size δx

Basically, the unit of calculations receiving input data, evaluating an output and possessing a transmission coefficient (trainable parameter) corresponds in this framework to one of the N^2 squared pixels of side δx contained in the plane with side $L = N \times \delta x$. The latter hosts an image, representing the intensity distribution of the electromagnetic radiation propagating through the network: that is the reason why, hereinafter, the neurons of diffractive layer will also be termed as "pixels". It is indeed well known that a pixel is the smallest unit of information composing a picture, therefore its identification with a neuron is the key ingredient to implement image classification through a neural network.

2.2.2 Inputs adjustment and forward propagation model

The mechanisms regulating the forward propagation of light are here presented into subdivided paragraphs, with the purpose of discussing the progressive interaction of light with the various diffractive layers, starting from the input and proceeding through the subsequent ones.

First of all, it is briefly introduced the notation used in the present section:

- x_i^0 represents the real value of amplitude of the pixel i in the input image, which, as will be clearer later, is directly projected onto the input layer
- $x_i^{l=1}$ represents the complex value of the neuron i in the input layer (identified as $l = 1$, with l indexing layers)
- y_i^l denotes the complex input to pixel i in layer l
- z_i^l is the complex output from neuron i of layer l
- $t_i^l = e^{j\phi_i^l}$ represents the transmission coefficient of neuron i at layer l; the spatial position (x_i, y_i, z_i) of Eq.2.1 is omitted to avoid heavy notation

It is emphasized that a complex field input to or output from a given layer is identified as an $N \times N$ matrix; each of its elements, labeled by the index $i \in [1, ..., N^2]$ with a precise

order maintained equal for any plane, represents the value of the field in correspondence of the i^{th} pixel (or neuron) of the considered layer.
Despite two indexes ($k = 1, ...N; \quad l = 1, ..., N$) constitute the proper and usual choice to label an element (at row k and column l) in a 2D matrix, here, apart from expressions of more complicated element-wise operations, an unique index $i = 1, ..., N^2$ will be used in order to highlight the univocal association of the matrix element with one of its N^2 pixels.
In the following, each above-mentioned $N \times N$ matrix will be denoted by a capital letter, the same as the lowercase one used to name its internal elements, which are the inputs or outputs at any single pixel. For instance, $X^{l=1}$ is the matrix representing the complex field output from the input layer and has internal elements denoted as $x_i^{l=1}$, while T^l is the matrix of trainable parameters $t_i^l = e^{j\phi_i^l}$ at the first hidden layer.
The connection between the matrix mathematical representation and the physical layer is identified by the finite size δx of each pixel, which, as shown in the following, will be involved in any calculation of the electromagnetic radiation - related physical quantities measurable over a layer.

Image at the input layer

The input handwritten digits provided by the MNIST database (Modified National Institute of Standards and Technology database) [1] contain of 28×28 pixels with grayscale values, from 0 (foreground, black) to 255 (background, white). As an example, an image from the training dataset is shown in Fig.2.3.

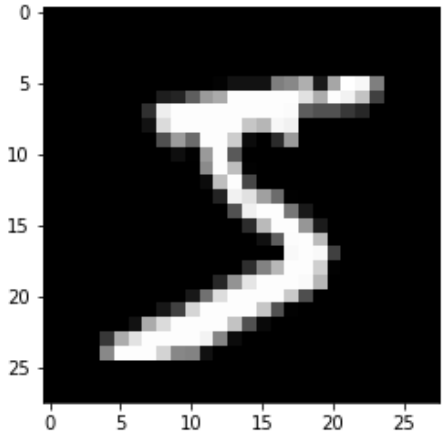


Figure 2.3: Handwritten digit from the MNIST dataset, with 28×28 pixels

[1] The dataset has been taken from American Census Bureau employees and high school students

The real value taken by a pixel quantifies the light intensity in that precise region of the image.
A laser beam of wavelength of λ, carrying information about the image to be classified, is projected onto the input layer and propagates through the network. The pixel i of the input layer ($l = 1$) has a complex value $x_i^{l=1}$ given by:

$$x_i^{l=1} = x_i^0 \times t_i^{l=1} = x_i^0 \times e^{j\phi_i^{l=1}} \tag{2.3}$$

where x_i^0 denotes the real-valued amplitude of the image to be classified in correspondence of the pixel i.
Using matrix notation, Eq.2.3 is written as:

$$X^{l=1} = X^0 \circ T^{l=1} \tag{2.4}$$

where "$\circ$" denotes the Hadamard (or equivalently entrywise) product [2].
Basically, in the present model no physical distance separates the plane displaying the input image and the input layer: it is considered a direct projection of light onto the network's first layer, whose pixels have complex-valued transmission coefficients with trainable phase ϕ.
It is thus clear that the input image must represent a distribution of light's wave amplitude that, once multiplied pixel by pixel to the complex exponential functions encoded in each neuron of the input layer, determines, in correspondence of the latter, a complex field $X^{l=1}$. The latter is necessarily (for numerical implementations) a discrete function assuming the complex value x_i^1 in correspondence of the neuron $i \in [1, ..., N^2]$ at position $\vec{r_i} = (x_i, y_i, z_i)$ and simply represents an approximation of the continuous complex field $U(x, y, z)$ introduced in section 1.2.1, which propagates through the network from one plane to the following one and is diffracted by each of them.
The relationship – shortly detailed – between the value of x_i^0 and of the i^{th} pixel in the input MNIST digit must clearly involve a square root operation, since mathematical analysis of electromagnetic waves' propagation deals with complex fields whose modulus squares represent light intensities (Eq.1.28 of paragraph 1.2.2).
Being the $N \times N$ matrix of complex values $X^{l=1}$ obtained through a Hadamard product (2.4), the input image and the input layer must have the same dimensions. As discussed in the next sections, diffractive layers will require to consist of more than $N^2 = 28 \times 28$, consequently an image resize will be necessary for the input image. The library of programming functions called "OpenCV" (Open source computer vision) provides various interpolation algorithms to resize images: in this specific case, the "nearest-neighbor interpolation" turns out to generate an image containing more pixels, without visible differences with respect to the original one.

[2]This binary operation takes two matrices of the same dimensions and generates a third matrix of the same dimension, where each element k, l (at row k and column l) is the product of the two elements k, l of the original two matrices

With the purpose of experimentally implementing the studied device based on free-space optical communication, a low "diffracted beam" divergence is important. The divergence of the propagating radiation with respect to the optical axes is controlled by measuring the value of light intensity permeating each plane: if such quantity is approximately constant from input to output of the network, the diffracted beam can be considered "sufficiently collimated". To set a reference value, it is established to normalize to 1 the intensity distribution "summed up" (if it was a continuous function, it would be integrated) over the input image.
The following operations on the input digits' pixels are thus performed: the intensity value I_i at each pixel $i = [1, ..., N^2]$ of every MNIST image – like the one of Fig.2.3 – is scaled down, square-rooted and the resulting amplitude A_i is finally normalized, leading to the value of x_i^0 appearing in 2.3:

$$I_i \Longrightarrow I_i' = \frac{I_i}{255} \Longrightarrow A_i = \sqrt{I_i'} \Longrightarrow x_i^0 = \frac{A_i}{\sqrt{\sum_{j=1}^{N^2} I_j' \times (\delta x)^2}} \tag{2.5}$$

In this way, the sum of light amplitudes extended to all pixels of each input image satisfies the following condition:

$$\sum_{i=1}^{N^2} |x_i^0|^2 (\delta x)^2 = 1 \tag{2.6}$$

Propagation from input to output layer

The propagation along the z direction of the electromagnetic radiation incident perpendicularly on a diffrative layer and propagating until the next one is implemented in the neural network by working in the frequency domain. The sequential operations described below, based on the theory introduced in Section 1.2.1, refer to information propagation between the input and the first hidden layer.

1. The complex field emerging from the input layer ($l = 1$) and encoded in the $N \times N$ matrix $X^{l=1}$ containing elements $x_i^{l=1}$ (2.3) labeled by $i = [1, ..., N^2]$ is 2D discrete Fourier transformed, giving as a result the matrix A_0, which represents a numerically implemented angular spectrum:

$$A_0 = \mathcal{F}\{X^{l=1}\} \tag{2.7}$$

2. The angular spectrum A_0 is propagated along the distance d between the two considered layers:

$$A_1 = A_0 \times e^{jk_z d} \tag{2.8}$$

where $k_z = \frac{2\pi}{\lambda}\sqrt{1 - \alpha^2 - \beta^2}$, with α and β terming the direction cosenes, according to the notation of Section 1.2.1

3. A 2D discrete Fourier antitransform is applied to A_1 to give as a result the complex electromagnetic field $Y^{l=2}$ input of the first hidden layer ($l = 2$):

$$Y^{l=2} = \mathcal{F}^{-1}\{A_1\} \tag{2.9}$$

4. The entrywise product is performed between $Y^{l=2}$ and the matrix of the transmission coefficients of the layer $l = 2$, $T^{l=2}$:

$$Z^{l=2} = Y^{l=2} \circ T^{l=2} \tag{2.10}$$

where the i^{th} element of $T^{l=2}$, given by $t_i^{l=2} = e^{j\phi_i^{l=2}}$, involves the trainable parameter $\phi_i^{l=2}$.

The matrix $Z^{l=2}$ obtained in Eq.2.10 represents the output from the first hidden layer and, if subjected to operations $1) \implies 2) \implies 3) \implies 4)$, is propagated until the subsequent plane. By iterating this procedure, the electromagnetic wave is propagated until the output layer.

The numerical operations presented above are now described with greater detail.

2D Discrete Fourier transform (2D DFT) The 2D Discrete Fourier transform (2D DFT), generally defined for complex inputs and outputs, can be defined in different ways, depending for example on the sign of the exponent, or on the used normalization. The function implemented in the code uses the 2D DFT defined as (referring to Eq.2.11: $A_0 = \mathcal{F}\{X^{l=1}\}$)

$$A_{0,ik} = \sum_{p=0}^{N-1} \sum_{q=0}^{N-1} X_{pq}^{l=1} \exp\left\{-2\pi j \left(\frac{pi}{N} + \frac{qk}{N}\right)\right\} \tag{2.11}$$

where

$$i = 0, \ldots, N-1; \quad k = 0, \ldots, N-1$$

label respectively the row and column of element $A_{0,ik}$.
The DFT is computed through a very fast algorithm, called "Fast Fourier Transform" (FFT), which has contributed to the current force of discrete Fourier transform in numerical computations.

In phase $1)$, it is actually applied, after performing Eq.2.11, a function called "Fast Fourier Transform shift" (fftshift), which rearranges the matrix A_0 by shifting its zero-frequency components to the center of the spectrum. In particular, the "Fast Fourier Transform shift" swaps the first quadrant of a matrix with the third, and the second quadrant with the fourth,

Discrete form of propagator e^{jk_zd} In order to numerically implement the propagator e^{jk_zd} that, applied to the angular spectrum A_0 at input layer layer, determines its propagation along distance d hence the angular spectrum $A_1 = A_0 \times e^{jk_zd}$ at the first hidden layer, two $N \times N$ matrixes F_x and F_y must be defined. Their elements correspond to the spatial frequencies defining the wave vector $\vec{k} = 2\pi(f_x, f_y, f_x)$ (Eq.1.12) and are thus used to calculate the propagator's exponent:

$$e^{jk_zd} = e^{j2\pi\sqrt{(\frac{1}{\lambda})^2 - F_x^2 - F_y^2}d} \tag{2.12}$$

As well-known, the spatial domain is the visible image space, where distances in the image (in pixels of size δx) correspond to real distances. Fig.2.4 represents, for the sake of clarity, the simple correspondence between 1D spatial domain *(a)* to 1D frequency domain *(b,c)*. The latter, allowing to perform more easily image processing's operations that would result complicated in the former, basically represents the value change rate of pixels.

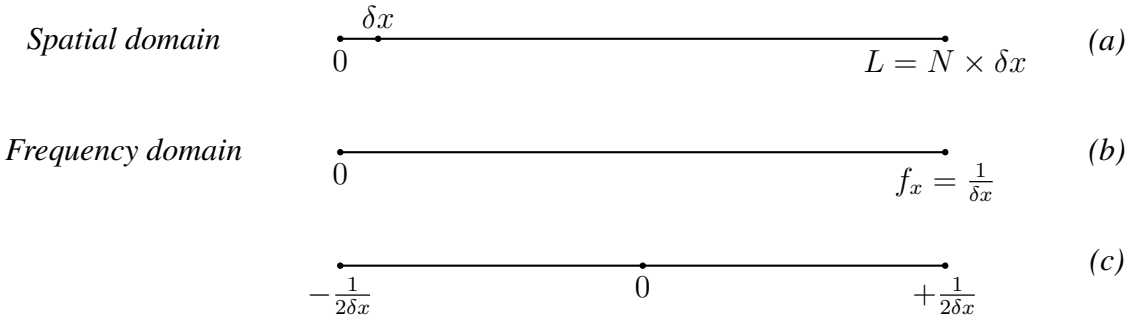


Figure 2.4: Correspondence between spatial *(a)* and frequency *(b,c)* domain

Considering to deal with a large enough number of pixels N in a squared image of size L, the spatial frequencies f_x and f_x are defined in the range of frequencies

$$\left[0, \frac{1}{\delta x}\right] = \left[0, \frac{N}{L}\right]$$

A frequency domain symmetrical with respect to zero-frequency *(c)* is taken, in order to satisfy the Nyquist–Shannon sampling theorem, fundamental when implementing discrete signals approximating continuous ones. It provides the following sufficient condition for sampling a continuous signal with no loss of information: for a given sample rate f_x, perfect signal reconstruction is ensured possible for a bandlimit $B < \frac{f_x}{2}$. If a higher bandlimit (Fig.2.4-(b)) is used, the image reconstruction (in real space) may exhibit "aliasing", namely distortions making it different from the proper one, due to the fact that the frequency signal have cut away fundamental information of the image,when

sampled.
To summarize, the two $N \times N$ matrixes F_x and F_y have respectively identical rows and identical columns of N values equally spaced in the range

$$[-\frac{1}{2\delta x}, +\frac{1}{2\delta x}] = [-\frac{N}{2L}, +\frac{N}{2L}]$$

By performing the operation "$-(F_x^2 + F_y^2)$" at the exponent of the expression 2.12, all the possible combinations between any two values of f_x and f_y are considered and the $N \times N$ matrix propagator $e^{jk_z d}$ can be element-wise multipled by the tensor A_0 to allow propagation of all its elements.

Output layer

The electromagnetic radiation transmitted by the last hidden layer (containing, like every layer apart from the output one, transmission coefficients with trainable phase) propagates along the distance d until the output layer. The latter contains $N \times N$ pixels and encodes the positions of 10 photodetectors arranged on it: pixels with value 1 and 0 represent respectively presence and absence of a detector in that specific area of the plane. Each photodetector is associated univocally to a digit and the system is trained to focus light on a specific detecting region of the output plane, according to the class of the input image. Basically, once training is complete, photodetectors are expected to light up when the digit they represent is shown to the machine.
Despite the extension of the detecting regions will be detailed in the next paragraph, it is anticipated that the total area occupied by photodetectors must occupy a suitable portion of the output plane to perform proper light detection. Furthermore, detectors are not in contact with each other and are symmetrically arranged to avoid any biased configurations, which may lead to lower system's performance if light is expected to illuminate the whole plane.
As an example, an output plane of 28×28 pixels containing 10 detecting regions of size 4×4 is shown in Fig.2.5: areas occupied and unoccupied by detectors are respectively identified by white pixels (with value 1), and black pixels (with value 0).

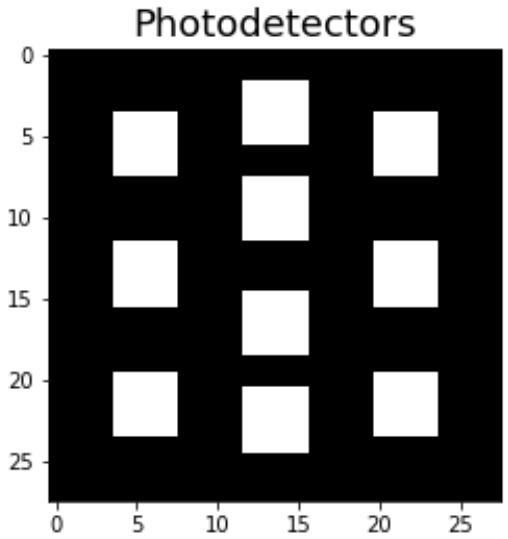


Figure 2.5: Photodetectors of size 4×4 in output plane with 28×28 pixels

Fig.2.6 highlights the specific positions of photodetectors associated to each digit, representing the different configurations of output plane depending on the provided input digit, in an ideal case of perfect learning.

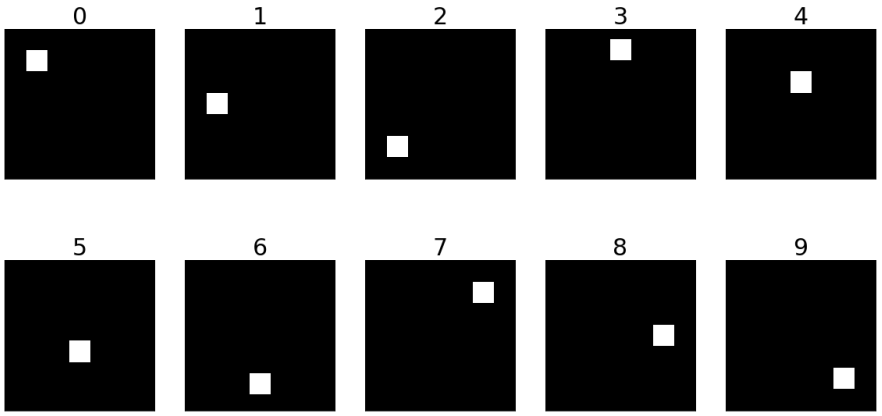


Figure 2.6: Single photodetectors' positions associated to a specific digit. Top left figure: after a proper training, the machine fed with a "0" digit is expected to focus light onto the upper-left corner of the output plane

.

The mathematical operations performed by the output layer are now detailed:

- It receives an input matrix $Y^l = Y^{OL}$ (where l=OL stands for "output layer") of dimension (N, N) encoding the complex values of electromagnetic field projected onto it
- It computes the matrix of electromagnetic field's intensity given as input to the output layer, Y'^{OL}. Its i^{th} element y_i^{OL} is given by

$$y_i'^{OL} = |y_i^{OL}|^2(\delta x)^2 \tag{2.13}$$

  where y_i^{OL} is the i^{th} element of matrix Y^{OL} and the constant $(\delta x)^2$ is introduced to physically associate light intensity to finite area of each pixel.
- It converts Y'^{OL} into a 1D tensor of dimension $(1, N^2)$ denoted as Y''^{OL}
- It produces the output matrix Z^{OL} through the following matrix multiplication:

$$Z^{OL} = Y''^{OL} \times D^T \tag{2.14}$$

  where D^T is the transpose of the matrix D which has dimension $(10, N^2)$. Each row of D with N^2 elements encodes the position in the output layer of one specific detector: pixels of the detecting region are identified by "1", all the others by "0".

  As an example, to the arbitrary output layer named as "OL" of dimension $(4,4)$ (shown in Fig.2.7, which highlights the order of pixel labeling) containing two detectors each formed by 2×2 pixels, the following matrix D of dimension $(2,4^2 = 16)$ is associated:

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

  The output Z^{OL} is thus an array of dimension $(1, \#detectors)$: each of its elements refers to a specific detector and represents the total light intensity projected onto that detector.

2.2.3 Error evaluation and backpropagation

The results of the output plane are now compared with the targets and the resulting errors are backpropagated.
Each single target T is computed as a canonical array of dimension $(1, \#detectors)$ – or equivalently $(1, \#classes)$ – with a "1" in correspondence of the i^{th} component, where $i \in [0,9]$ is the label provided by the MNIST dataset together with the corresponding

ol_1	ol_2	ol_3	ol_4
ol_5	ol_6	ol_7	ol_8
ol_9	ol_{10}	ol_{13}	ol_{15}
ol_{11}	ol_{12}	ol_{14}	ol_{16}

Figure 2.7: Output layer "OL" of 4×4 pixels with two detectors, each marked by yellow regions of 2×2 pixels. It is highlighted the order of labeling pixels, by naming the i^{th} element of the layer pixel as "iol_i".

image to be classified.
The output array Z^{OL} is typically transformed into the normalized vector Z_N^{OL}, whose element in position $i \in [0,9]$ component is computed as:

$$Z_{N,i}^{OL} = \frac{Z_i^{OL}}{\sum_{i=0}^{9} Z_i^{OL}} \tag{2.15}$$

In this way all the components of Z^{OL} sum up to 1, as with T. The word "typically" of some lines above derives from the fact that some simulations without normalization have also been implemented, due to reasons described in the following chapter. For the moment, it is highlighted that *in this specific framework* normalization is not necessary to obtain output array's values in the range $[0,1]$, since the initial normalization performed in the input image (Eq.2.6) leads to $Z_i^{OL} < 1$ for each i. Nevertheless, normalization is generally applied to neural networks' output arrays when associations with canonical vectors have to be performed.
The comparison between Z_N^{OL} and T can be performed through the squared error:

$$E_{digit} = \frac{1}{10} \sum_{i=0}^{9} (Z_{N,i}^{OL} - T_i)^2 \tag{2.16}$$

Actually, the exact error which is backpropagated is not the one reported in Eq.2.16 which refers to a single image (hence the name "E_{digit}").
In fact, before training the training dataset is divided into an established number of parts, termed as "batches", since it is not possible to pass the entire database to the neural network at once. Despite the previous description concerned, for the sake of simplicity, the output produced by the system fed with a single image, the neural network is instead fed with all the images of a batch together. More specifically, the input is provided to the system as a tensor of dimension (BS, N^2), where "BS" denotes the "batch size", i.e. the cardinality of the batch; proper computations lead easily right back to the situation

described in the previous paragraph.
Consequently, a mean square error E_{batch} over the batch size is computed:

$$E_{batch} = \frac{1}{BS} \sum_{j=1}^{BS} E_{digit,j} \tag{2.17}$$

Finally, all the E_{batch} of all batches are averaged, giving the error of an epoch (defined shortly) E as a result:

$$E = \frac{1}{\#batches} \sum_{k=1}^{\#batches} E_{batch,k} \tag{2.18}$$

E represents the error to be backpropagated through the procedure described 1.1.2, where the stochastic gradient descent algorithm is automatically implemented by Tensorflow.
An epoch corresponds to the passage of the entire dataset to the neural network only once: during training, more epochs are settled, since the full dataset has to be passed multiple times to the machine in order to make it learn. As seen in the literature, the minimal number of epochs results generally to be 5, but it clearly depends on the implemented network.
Training duration is properly established by measuring at each epoch some quantities measuring the quality of learning, including the accuracy. This is used in our diffractive neural network and is defined as the percentage of correct classifications. Being evaluated at each epoch, it is given by the average of all the batches' accuracies. In the following, the accuracy evaluated at the last epoch in training phase will be denoted as α_{TR}, while α_{TE} will denote the same quantity evaluated for the test phase.

2.3 Setting of physical parameters and results

The physical parameters of the structure have to be properly setted to both match the condition of numerically simulated full connectivity and allow the experimental use of the network after its fabrication. Simultaneously satisfying the two requirements, so as to obtain good performances of the deep learning model together with robust conditions for its physical implementation, proves challenging. This is why finding a good compromise between the two constitutes the core of the present study, riddled with progressive modifications in both the the physical parameters and the structure of the network, in a sort of "trial" and error" approach followed on the basis of sometimes unpredictable simulation results.

2.3.1 Full connectivity and physical implementation's requirements

As is now known, the considered neural network uses optical diffraction to connect the neurons at various layers and, on the basis of diffraction notions reported in the Appendix,

the maximum half-cone diffraction angle θ satisfies

$$sin(\theta) = \frac{\lambda}{\delta x} \tag{2.19}$$

where δx is the layer feature size.

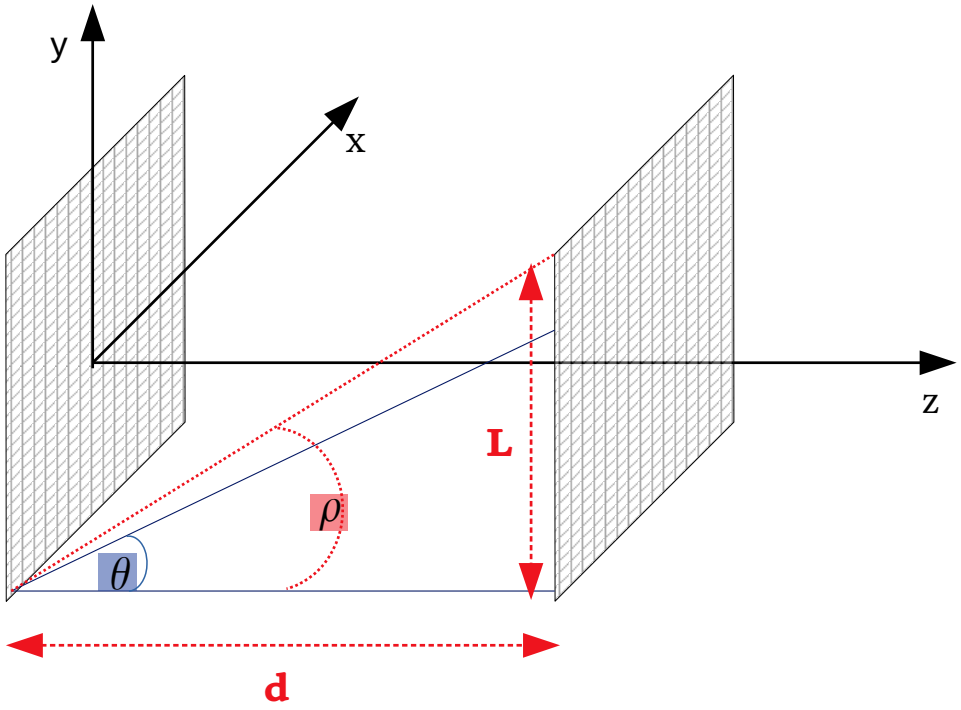


Figure 2.8: Optical diffraction connecting one neuron of a layer to neurons of the successive plane; the planes have side L and are placed at distance d from each other. The half-cone diffraction angle is represented by θ, while $\rho = arctan(\frac{L}{d})$.

In order to ensure full connectivity of the network, the distance d between two consecutive layers of side L must be relatively large, so that light diffracted from any pixel of a layer spans over the whole following plane at a distance d. Considering a section of the solid diffraction angle, as shown in Fig.2.8, the angle associated with radiation transmitted from a neuron placed on the corner of the left-layer is the one requiring the largest lower limit for full connectivity, namely:

$$\theta \geq \rho \tag{2.20}$$

where $\rho = arctan(\frac{L}{d})$.
The inequality 2.20 can be equivalently expressed as

$$tan(\theta) \geq \frac{L}{d} \tag{2.21}$$

or

$$tan\left[sin^{-1}\left(\frac{\lambda}{\delta x}\right)\right] \geq \frac{L}{d} \tag{2.22}$$

with clearly $\theta, \rho \in [0, \frac{\pi}{2}]$.

Starting from the above reported requirements, it is possible to explore appropriate values of parameters, basing on preconditions established by the practical implementation of the network.
First of all, an absolutely fixed physical parameter is the wavelength λ of the green laser pointer, available in the laboratory, emitting at $\lambda = 532nm$. The use of this relatively small wavelength radiation requires a small pixel size δx, in order to allow significant neurons' connectivity, as can be noticed by Eq.2.22.
On the other hand, even though submicrometric features are possible in the technique used to fabricate the layers, two-photon lithography, a minimal size δx_{min} of pixels needs to be established, since layers must be extended enough so that they can be handled, once fabricated, in laboratory. In fact, their side L cannot be made arbitrarily large by increasing N, due to unavoidable time-consuming numerical calculations when $N \geq 100{,}200$.
More specifically, the following range of minimal layer side L_{min} has been established

$$L_{min} \in [1mm, 2mm] \tag{2.23}$$

and, by imposing a maximum number of pixels per layer's side

$$N_{max} \in [100{,}200] \tag{2.24}$$

the value of δx_{min} is set to:

$$\delta x_{min} = 5um \tag{2.25}$$

Furthermore, it is not possible to arbitrarily decrease the quantity $\frac{L}{d}$ of Eq.2.22, so as to compensate a relatively large pixel size δx which would exactly make requirements 2.23 and 2.24 amply satisfied. This is due to the fact that the distance d between two adjacent layers cannot be significantly larger than their dimensions, otherwise their practical alignment, to be performed through proper optical systems employed in the laboratory setup, would be unfeasible.
With the premise that design parameters guaranteeing conditions for feasible and practical laboratory work can be estimated rather than firmly established a priori, the maximum value of d_{max} in an experimental implementation with layers of side L_{min} is approximated as:

$$d_{max} = 1cm \tag{2.26}$$

The total number of layers is set to be equal to three, including two diffractive planes with trainable parameters (input layer and one hidden layer) and the output plane. Aiming at

physically implementing a network with effective performances as close as possible to the numerically explored ones, a moderate number of layers is expected to guarantee minimal error sources in the experimental procedure. In fact, more hidden layers would clearly prove more challenging to be perfectly aligned between them, most probably leading to a configuration relevantly different with respect to the trained one, which is instead intended to establish the design of a physical device performing the desired task.
Actually, simulations with more diffractive layers, up to 5, have been performed to explore the impact of the number of layers on the machine performance, where the latter is expected to increase with the former for previous considerations on deep learning models.
As will become clearer later, another important reason for training a machine with different physical parameters, which do not match the requirements settled by our laboratory setup, stands in the comparison of performances achieved by the diffractive deep neural network introduced in [18] by X.Lin et al. They realized a 5 hidden layers- diffractive framework, using an illumination system with wavelength of $0.75mm$ [3] in air, layers outdistanced of $3cm$ and of area of $8cm \times 8cm$, covered by 200×200 pixels of size $400um$.

It is emphasized that, in all the simulations presented in the next section, the sum over all pixels of light intensity Y'^{OL} given as input to the output layer (whose components $y_i'^{OL}$ are defined in Eq.2.13) satisfies the following equation:

$$\sum_{i=1}^{N^2} |y_i'^{OL}|^2 (\delta x)^2 = 1 \tag{2.27}$$

and the same holds for the hidden layer/layers, meaning that electromagnetic energy is conserved over each plane, since it proves to be unchanged with respect to the one extended at the input layer, determined by Eq.2.6 after the normalization performed in Eq.2.5.
Energy conservation in correspondence of the different layers arranged along the optical axes allows to describe the radiation incident on them as a *paraxial ray*, which is defined as a ray forming a small angle with respect to the optical axes and, consequently, lying close to it during propagation.
In optics, when such condition is satisfied, the so called "paraxial approximation" allows to significantly simplify calculations of the wave propagation. It involves, for the wave vector $\vec{k} = \frac{2\pi}{\lambda}(\alpha, \beta, \gamma) = 2\pi(f_x, f_y, f_z)$ represented in Fig.1.6 (α, β, γ are the direction cosenes) and the angle $C = cos^{-1}(\gamma)$ spanned by it with respect to the optical axes z, the following approximations:

$$sin(C) \approx C \qquad tan(C) \approx C \qquad cos(C) \approx 1 \qquad |k| \approx |k_z| \tag{2.28}$$

[3] According to previous considerations (Eq.2.22), a significant larger wavelength allows the use of noticeably wider layer feature sizes δx and clearly results in larger dimensions of the device.

In the neural network's simulations presented in the following, the third condition of Eq.2.28 can be indeed directly verified by evaluating

$$cos(C) = \gamma = \sqrt{1-\alpha^2-\beta^2} = \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2} = \sqrt{1-(\lambda f_x)^2-(\lambda f_y)^2}$$

which, after replacing the maximal value of spatial frequencies $f_x, f_y \in [-\frac{1}{2\delta_x}, +\frac{1}{2\delta_x}]$ in it, will lead to the following relation for any later reported configuration, which has been numerically trained:

$$cos(C) = \sqrt{1-\left(\frac{\lambda}{2\delta_x}\right)^2-\left(\frac{\lambda}{2\delta_x}\right)^2} = \sqrt{1-\left(\frac{\lambda N}{2L}\right)^2-\left(\frac{\lambda N}{2L}\right)^2} \approx 1 \qquad (2.29)$$

Now, some of the relevant results which have charted the way to phase masks of 200×200 pixels, considered suitable for diffractive layers fabrication, are reported.
Unless specified differently, the error calculation is assumed to be performed starting from a normalized output vector Z_N^{OL}, whose components $Z_{N,i}^{OL}$ are obtained through Eq.2.15. Moreover, the value of batch size will be setted to $BS = 8$ in all the training simulations.

2.3.2 Layers with 28×28 pixels

In deep learning models performing image classification, neural networks with layers of few pixels are clearly not expected to perform as good as the ones involving more neurons in propagation information.
In the present framework, accuracy α_{tr} larger than 80% can be obtained when planes with a relatively small number of pixels, 28×28, are employed. Nonetheless, as described below, such configurations cannot satisfy the requirements described in Paragraph 2.3.1.

Laser source with wavelength $\lambda = 0.75mm$

If larger wavelengths than the one of interest are employed, pixels can clearly increase in dimension. It is here considered the case of $\lambda = 0.75mm$, both to appreciate the change in physical parameter values (with respect to a network supplied by laser of $\lambda = 532nm$) and and to provide a comparison with the framework developed in [18] by X Lin et al. Actually, their structure slightly differs from ours in some configuration features and as specified above it employs $N^2 = 200 \times 200$ pixels per layer. However, simulations with $\lambda = 0.75mm$ and relatively few neurons have represented a timesaving starting point to verify the reliability of our neural network. The obtained classification accuracy larger than 80% can in fact be considered promising, by comparison with the values of α_{tr} equal to 55.64% and 91.75% presented in [18] for respectively one and five diffractive layers containing much more neurons. Training later performed with $N = 200$ and exactly the

same parameters used by X Lin et al. has lead to $\alpha_{tr} = 74.70\%$ in our framework with two diffractive layers.
It is emphasized that accuracies of approximately 80% are generally (at any value N of pixel involved) considered good in our framework, due to – as already highlighted – the use of linear representation patterns, totally unusual in deep learning models and resulting from the use of light diffraction for information propagation.

In the case $N = 28$, the distance between layers, their size and pixels' dimensions are respectively given by

$$d = 5.0cm \quad L = 2.4cm \quad \delta_x \simeq 0.86mm$$

A glimpse of the 10000 outputs given by the system during the test phase, after training performed within 5 epochs, is provided in Fig.2.9 (where, as in the following images, layers' regions of maximal and minimal light intensity appear respectively as yellow and violet, after colors adaptation to better highlight bright pixels with respect to dark ones). Here, in correspondence of each input digit, an output and a masked output distribution are represented: the former identifies the light intensity incident on the output plane, defined as $|Y^{OL}|^2$ in Paragraph 2.2.2; the latter reproduces exactly the output distribution resulting from the application of a detectors mask. While such mask is not employed during training since just the real intensity $|Y^{OL}|^2$ must be involved in error evaluation, it could instead be aligned to the output layer during the experimental phase of network's physical implementation. As shown below, the masked output distributions exhibit a clearer lighting of each photodetector in response of a given input digit (all the specific correspondences are illustrated in Fig.2.5).

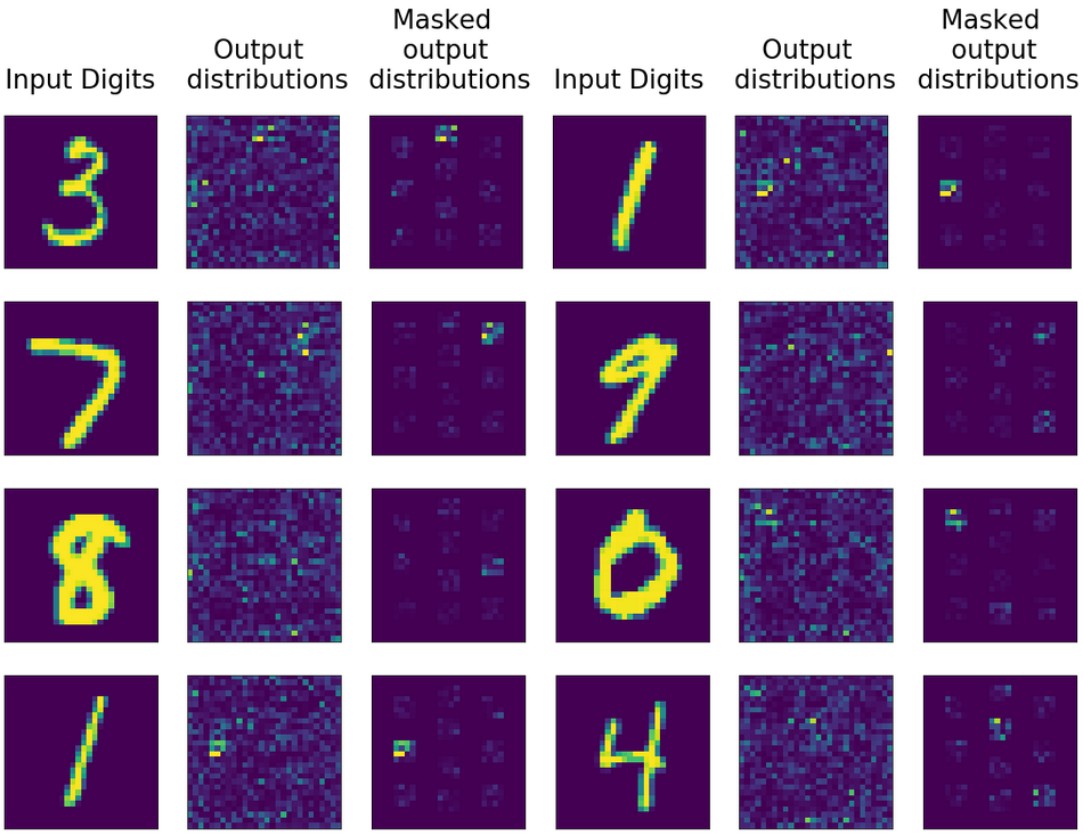


Figure 2.9: Data from test set; network with two layers of 28×28 pixels, $\delta_x \simeq 0.86mm$, $L = 2.4cm$, $d = 5.0cm$; training performed with 5 epochs, batch size=8.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution incident on the output plane after a detectors mask has been applied.

All the labels reported Fig.2.9 are predicted correctly by the network, even the "9" in the second image from the top, on the right, which appears as almost mistaken for a "7" form the masked output distribution.

Visualization of the network's performance is allowed by the confusion matrix, shown here below, displaying the ways in which the classification model is confused when it makes predictions. Each row of the table corresponds to an actual class, each column to a predicted class and its elements represent the counts of correct and incorrect classifications.

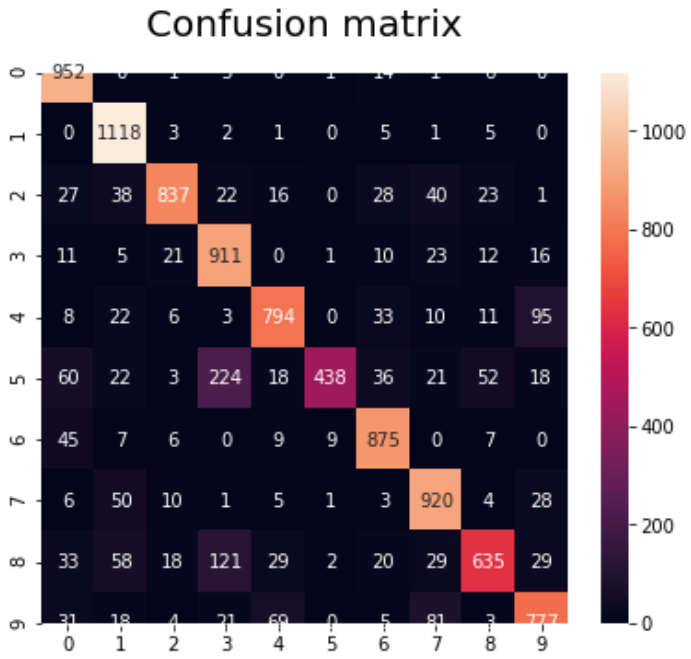


Figure 2.10: Confusion matrix; network with two layers of 28×28 pixels, $\delta_x \simeq 0.86mm$, $L = 2.4cm$, $d = 5.0cm$; training performed with 5 epochs, batch size=8

The classification accuracy α_{tr} is reported as a function of the epoch number in Fig.2.11, where it shows to reach a maximum value approximated as 81.26% in the last epoch of the training phase.

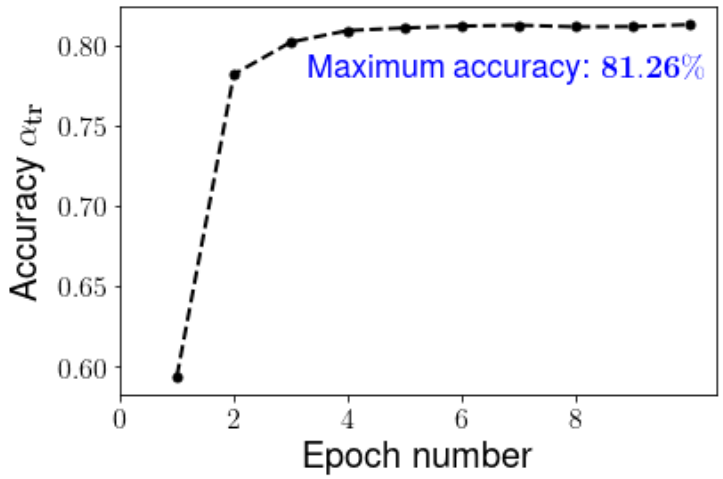


Figure 2.11: Classification accuracy α_{tr} as a function of the epoch number; network with two layers of 28×28 pixels, $\delta_x \simeq 0.86mm$, $L = 2.4cm$, $d = 5.0cm$

Laser source with wavelength $\lambda = 532nm$

Focusing now on the radiation's wavelength of interest, $532nm$, the following considerations highlight how unfeasible would be to physically implement a neural network with layers of $N = 28$. Using pixels of size $\delta x = 10um$ would still lead to a relatively moderate ratio $\frac{L}{d} \approx 0.05$, but with absolutely extremely small hence not handable objects of size $L = 0.28mm$. By increasing the value of δx, it turns out that exploitable layers with size of the order of few millimeters should be spaced from each other by a relatively enormous distance d of several centimeters, in order to satisfy the conditions presented in 2.3.1. For instance, layers with side $L = 1.4mm$ containing pixels of size $\delta x = 50um$ would need to be outdistanced of $d \geq 13cm$.

Layers of side $L = 0.28mm$ outdistanced by $d = 8cm$ Described the incompatibility of $N = 28$ and $\lambda = 532nm$ for experimental realization, the present paragraph is actually dedicated to analyze some "unrealizable" diffractive networks implemented with the above parameters. In particular, layers of side $L = 0.28mm$, containing pixels of dimension $\delta x = 10um$ and outdistanced of $d = 8cm$ result in a physically unreal framework which, by largely satisfying the requirement of full connectivity, reaches in specific geometrical conditions of the output plane a good classification accuracy over the training set and can be used to accomplish fast (because of few pixels) and reliable studies on the detectors' size.
As shown in Fig.2.12, reporting data extracted from the test phase, the unmasked output distribution of a network achieving a satisfying classification accuracy $\alpha_{tr} \approx 80.78\%$ and with detectors of 4×4 pixels does not prove that the system has correctly classified the input image. In fact, apart from digit "3" and "1" determining a visible lighting of the corresponding detector, all the other input images result in an ambiguous "output distribution" of light intensity and would require to pose the detector mask onto the output plane in order to be properly classified in the laboratory.

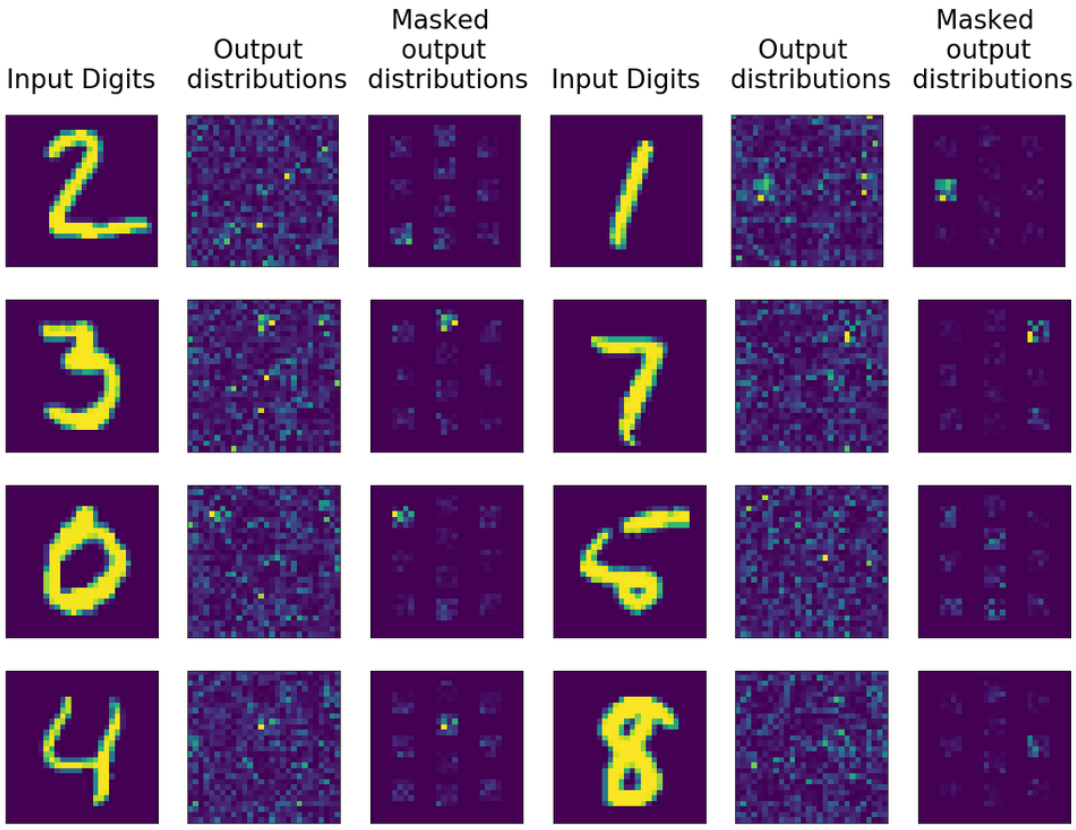


Figure 2.12: Data from test set, referred to a network with two layers with 28×28 pixels, $\delta x = 10um$, $z = 8cm$, detectors of 4×4 pixels.
Error E is evaluated through a normalized output error. Training performed with 5 epochs and batches of $BS = 8$ gives a classification accuracy $\alpha_{tr} \approx 80.78\%$.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied.

Incidence of light onto non detecting regions is due to the fact that during learning the error E is minimized by attempting to maximize (as far as possible) the element in position i of the normalized output array Z_n^{OL} – defined in Eq.2.15 – when digit $i \in [0,9]$ is given as an input. Basically, transmission coefficients of diffractive layers are modified to make light focus more on the correct photodector with respect to the other nine detecting regions of the output plane, but not with respect to all the rest of the output plane.
Output plane configurations with $\frac{1}{4} - \frac{1}{5}$ of the total area covered by detectors – as the ones illustrated in Fig.2.12 – turn out to produce good performances in deep learning models of image classification's literature.

In order to minimize light intensity distribution in non detecting regions while maintaining the same layout of the output plane, the following modification in the learning algorithm is applied: the output array Z^{OL} is no more normalized, so that it represent the *effective* light distribution in each photodetector and no more, like Z_n^{OL} does instead, the *relative* intensity in each 4×4 pixels' area with respect the other detecting regions. In this way, the maximization of an element of Z^{OL} should involve light intensity's increase in the corresponding detecting region and its simultaneous decrease in all the rest of the output plane. The modified learning model's outputs, associated to the same input images as Fig.2.12, are represented in in Fig. 2.13: one can notice that indeed output distributions, not significantly different from masked ones, exhibit light mostly focusing on detection regions and consequently prove sufficient to deduce the system's predictions. Nonetheless, the neural network does not show good performances, as is suggested by the relative small value of classification accuracy $\alpha_{tr} \approx 61.07\%$. Even though few examples are obviously not meaningful to represent the system's performance (as reported at the bottom of Fig.2.13, the system does in fact correctly predict all the eight digits of the figure, in spite of an accuracy of only 61.07%), it is clear that, apart from visible recognition accomplished by the system for digit "4", "1", "7" [4] involving prevailing lighting of a single detector over all the others, for the remaining input images it would not be possible to deduce the "predicted labels" with sufficient certainty, by simply looking either at the output or at the masked output distributions.
This is absolutely not the case of Fig.2.12, where the observer may easily guess machine predictions reported at the bottom after a quick look to the masked output distributions, probably except for the one referred to an "insidious" digit "5", whose challenging classification for the network results indeed in a wrong prediction.

[4] Digits "1" and "7" are, in fact, among the ones best classified after any training process

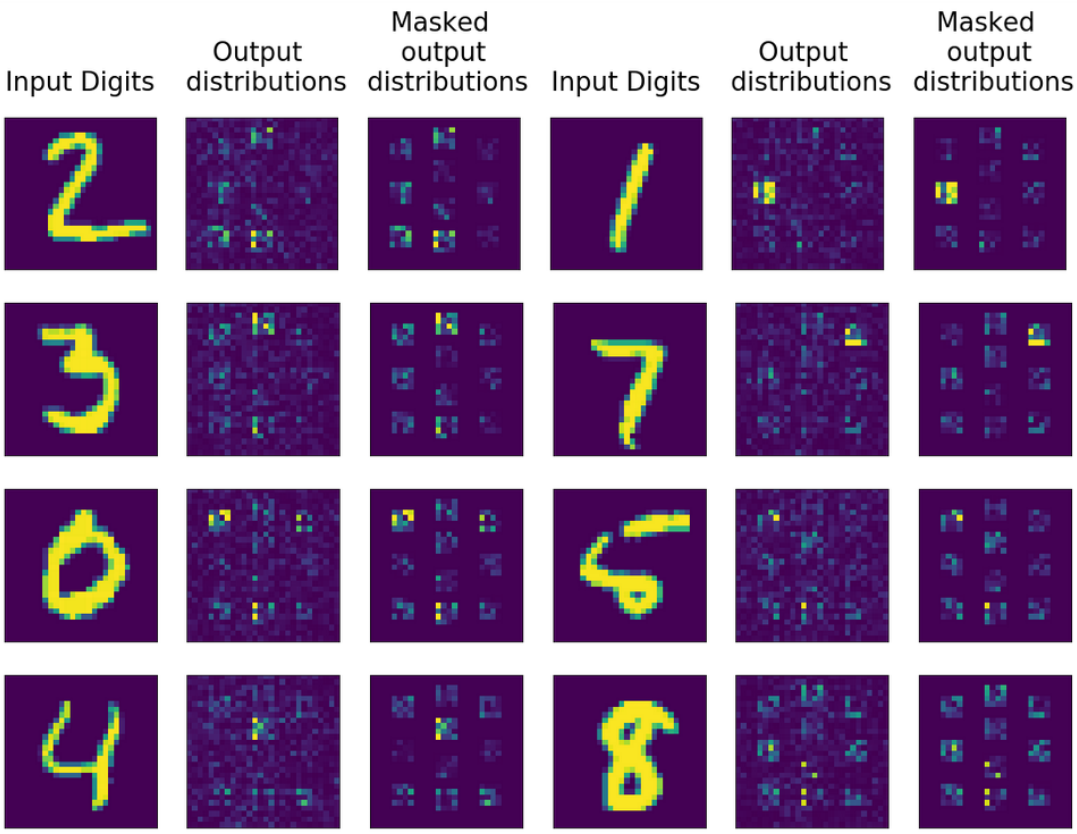


Figure 2.13: Data from test set, referred to a network with two layers with 28×28 pixels, $\delta x = 10um$, $d = 8.0cm$, detectors of 4×4 pixels.
Error E is evaluated through a not normalized output error. Training performed with 5 epochs and batches of $BS = 8$ gives a classification accuracy $\alpha_{tr} \approx 61.07\%$.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied.

The "confusion" of the system emerging from Fig.2.13 is explained by the lack of normalization of the output array Z^{OL}. Unnormalized output vectors present in fact, at the end of training phase, a maximum element which is more "distant" from 1 with respect the highest element of normalized arrays Z_n^{OL}. The machine is basically unable to adequately modify its internal trainable parameters so as to properly arrange values in the output obtain and make it sufficiently close to the target one, thus it does not achieve sufficient learning.

A different solution to minimize light intensity in non detecting regions of the output plane could be to increase photodetectors' size. Fig.2.14 shows the outputs obtained with a network using the same physical parameters as before, except for using detectors of 7×7 pixels, when error calculation is performed with a normalized output vector Z_N^{OL}.With respect to Fig.2.12, the output distributions are more similar to the masked output ones; nonetheless, now the latter exhibit just a faint highlighting of the correct detector, except for some cases of input digits (including "1" and "7", which, as previously said, are commonly the most correctly classified).

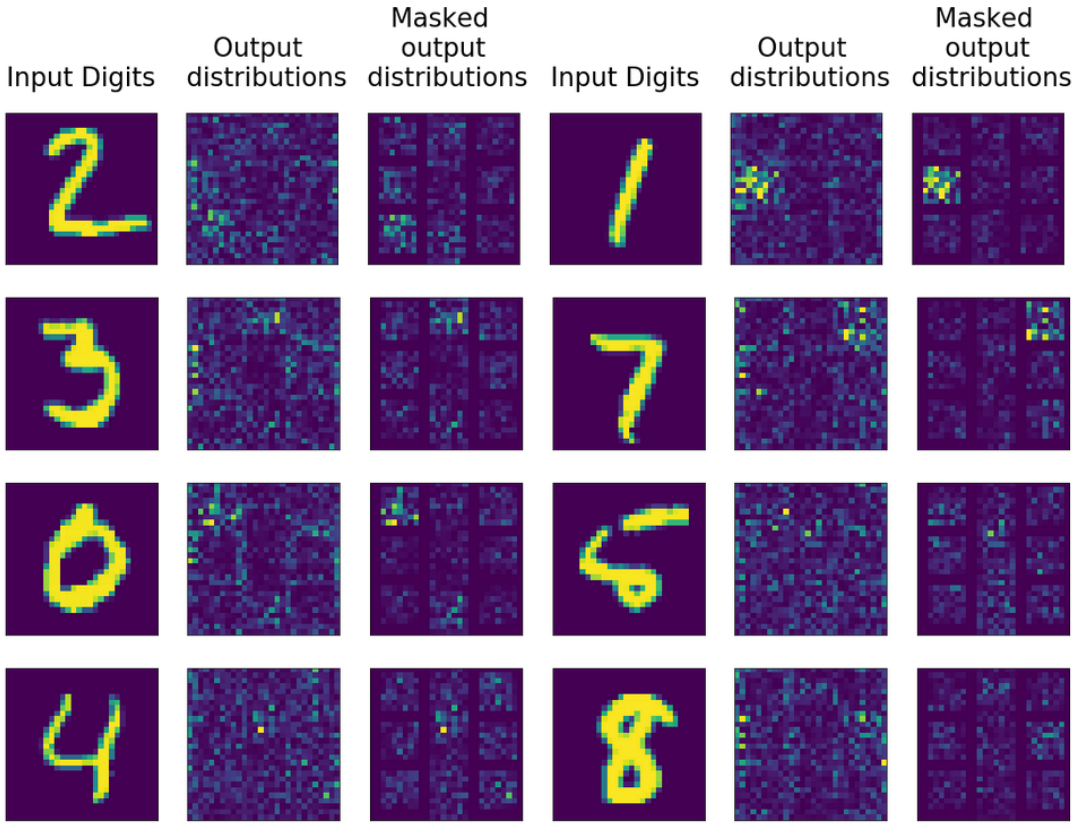
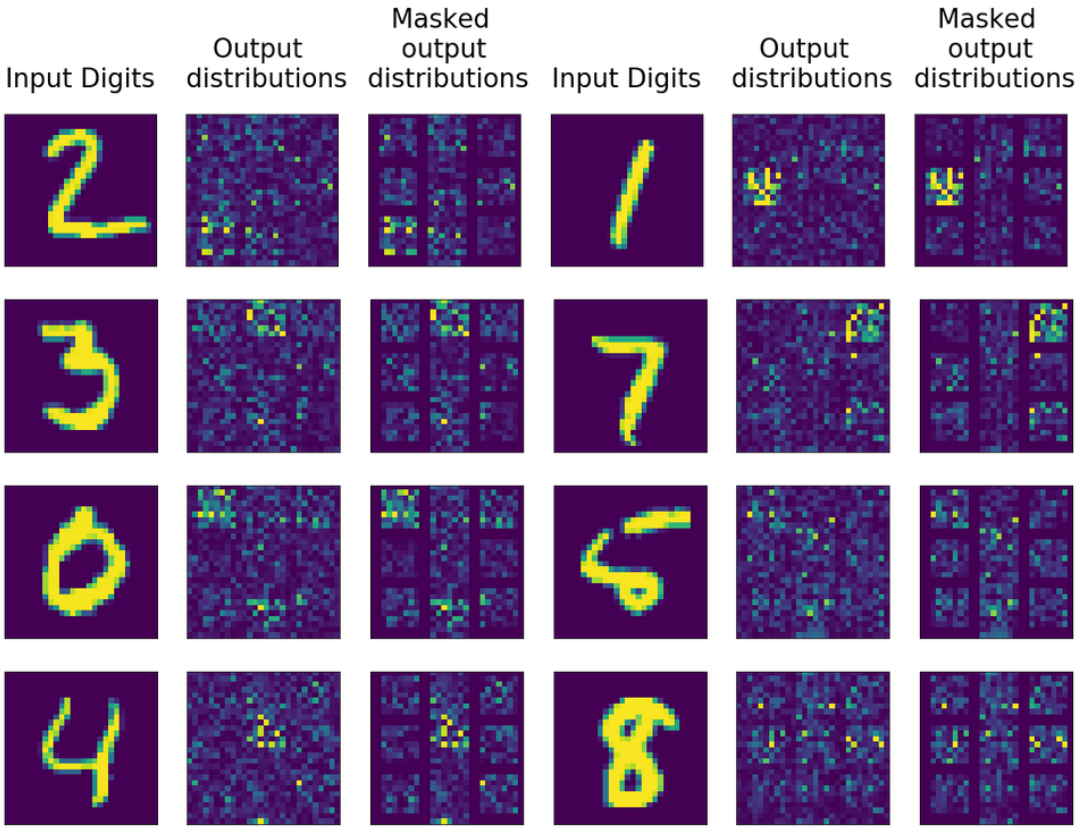


Figure 2.14: Data from test set, referred to a network with two layers with 28×28 pixels, $\delta x = 10um$, $d = 8cm$, detectors of 7×7 pixels.
Error E is evaluated through a normalized output error. Training performed with 5 epochs and batches of $BS = 8$ gives a classification accuracy $\alpha_{tr} \approx 74.99\%$.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied.

A decreasing of training classification accuracy from approximately 80.78% to $\alpha_{tr} \approx 74.99\%$ proves the worsening of performances. Making detectors extremely closer to each other leads to lower accuracy in a system that on average correctly addresses light onto a single detecting region.

When employing an unnormalized output vector in error backpropagation, an increase in the value of α_{tr} from 61.07% to 67.86% is instead measured when passing from a network with 4×4 detectors to one with 7×7 detectors in the output layer. In Fig.2.15, lighting of the correct detector appears clearer when observing the output distributions (which are not significantly different from the masked ones, as expected with an unnormalized vector) with respect to Fig.2.13. This could be due to the fact that an increase of detectors size, leading to summing up more intensity contributions to each value of the output array, determines a relevant enhancement in the distribution associated to the correct detecting area, which therefore results closer to value "1" of the target array; in this way better learning can be performed in the same number of epochs.



True labels:	2	3	0	4	1	7	5	8
Predicted labels:	2	3	0	4	1	7	0	4

Figure 2.15: Data from test set, referred to a network with two layers with 28×28 pixels, $\delta x = 10um$, $d = 8cm$, detectors of 7×7 pixels.
Error E is evaluated through a not normalized output error. Training performed with 5 epochs and batches of $BS = 8$ gives a classification accuracy $\alpha_{tr} \approx 67.86\%$.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied.

In summary, introducing larger detectors leads, when error is evaluated through a normalized output vector, to a not negligible increase in accuracy, nonetheless this quantity maintains a relatively small value in the view of achieving good learning. Classification accuracy is instead generally higher when a normalized output array is employed and the passage from small to relatively large detectors even involves its decrease. Despite the lack of normalization makes light spread less out of detectors, more importance may be given to the network's accuracy; thus it is established to use photodetectors occupying $\frac{1}{4} - \frac{1}{5}$ of the total output plane's area, keeping into account to use a masking layer during experimental phase of physical implementation of the framework.

2.3.3 Layers with 100×100 pixels

As previously described, it is necessary to increase the number N^2 of pixels in each diffractive layer in order to approach to satisfy full connectivity with potentially exploitable geometrical parameters.
With $N = 100$ neurons, setting $\delta x = 10um$ would involve a minimal value d_{min} of distance between consecutive layers, given by (according to what seen in Paragraph 2.3.1):

$$d \geq \frac{N\delta x}{tan[sen^{-1}(\frac{\lambda}{\delta x})]} \iff d \geq d_{min}; \qquad d_{min} = 1.8cm$$

which still turns out to be excessively large with respect to the layer's side $L = 1mm$. As already mentioned, a compromise between full connectivity and possible experimental use of the network after its fabrication must be found out; therefore, simulations with $d < d_{min}$ are explored. They do not show good performances.
With the increase of pixel size δx, larger ratios $\frac{d}{L}$ are required, hence "making compromises" becomes more challenging: the distance between planes must in fact be setted to $d << d_{min}$ in order to obtain an experimentally exploitable framework, resulting in noticeably worse accuracies with respect to the ones obtained with $\delta x = 10um$. The resulting classification accuracies are summarized in the table below:

$L[mm]$	$\delta x[\mu m]$	$d[cm]$	α_{tr}	α_{te}
1.0	10.0	0.5	50.0%	65.5%
1.0	10.0	0.6	72.1%	62.5%
1.0	10.0	0.8	75.0%	50.0%
1.0	10.0	1.0	77.6%	62.5%
2.0	20.0	0.4	34.1%	10.0%
2.0	20.0	0.6	40.2%	15.0%
2.0	20.0	0.7	42.8%	15.0%
2.0	20.0	1.8	43.9%	25.0%
2.0	20.0	0.9	47.1%	12.5%
2.0	20.0	1.0	50.1%	25.0%

Table 2.1: Values of α_{tr} and α_{te} and the corresponding geometrical parameters, referred to a neural network with 2 layers of 100×100 pixels; training is performed with 5 epochs and $BS = 8$.

2.3.4 Layers with 200×200 pixels

By increasing number of pixels in a layer to $N^2 = 200 \times 200$ and considering $\delta x = 5um$, a minimal distance between layers $d_{min} \approx 0.93cm$ is required. Being the layer side

given by $L = 1mm$, a moderate ratio $\frac{d}{L}$ suggests the possibility of a good compromise between full connectivity and possible network's handling in experimental phase. The most relevant results are reported in the table below:

$L[mm]$	$\delta x[\mu m]$	$d[cm]$	α_{tr}	α_{te}
1.0	5.0	0.5	79.2%	62.5%
1.0	5.0	1.0	78.9%	50.0%
2.0	10.0	0.5	52.2%	25.0%

Table 2.2: Values of α_{tr} and α_{te} and the corresponding geometrical parameters, referred to a neural network with 2 layers of 200×200 pixels; training is performed with 5 epochs and $BS = 8$.

As shown in Table 2.2, increasing the pixel size leads to worse performances for the same reason described in Paragraph 2.3.3.
The best classification accuracy is even obtained for the network with geometrical parameters basing the design of the most easy to handle structure, namely layer side $L = 1mm$ and distance between planes $d = 0.5cm$. The corresponding test accuracy shows a moderate value; a relevant difference between α_{tr} and α_{te} is nevertheless common in deep learning models, since the test set is clearly formed by examples never presented to the machine during training.
Some examples of output distributions are provided in Fig.2.16, where it is possible to appreciate a sufficiently net lighting of the correct detector when digits "0", "2", "1", "6" are given as an input; a moderately ambiguous light intensity appears instead on the output plane in correspondence of input "9".
Though the use of a detectors mask layer would still improve the "reading" of systems' predictions in laboratory experiment due to intensity distribution still present in non detecting areas, it is possible to observe lack of light in nine out of ten detecting regions on the unmasked output layer, resulting in a sort of projection of nine "shadows" tracing shapes of the photodetectors not associated to the input digit.

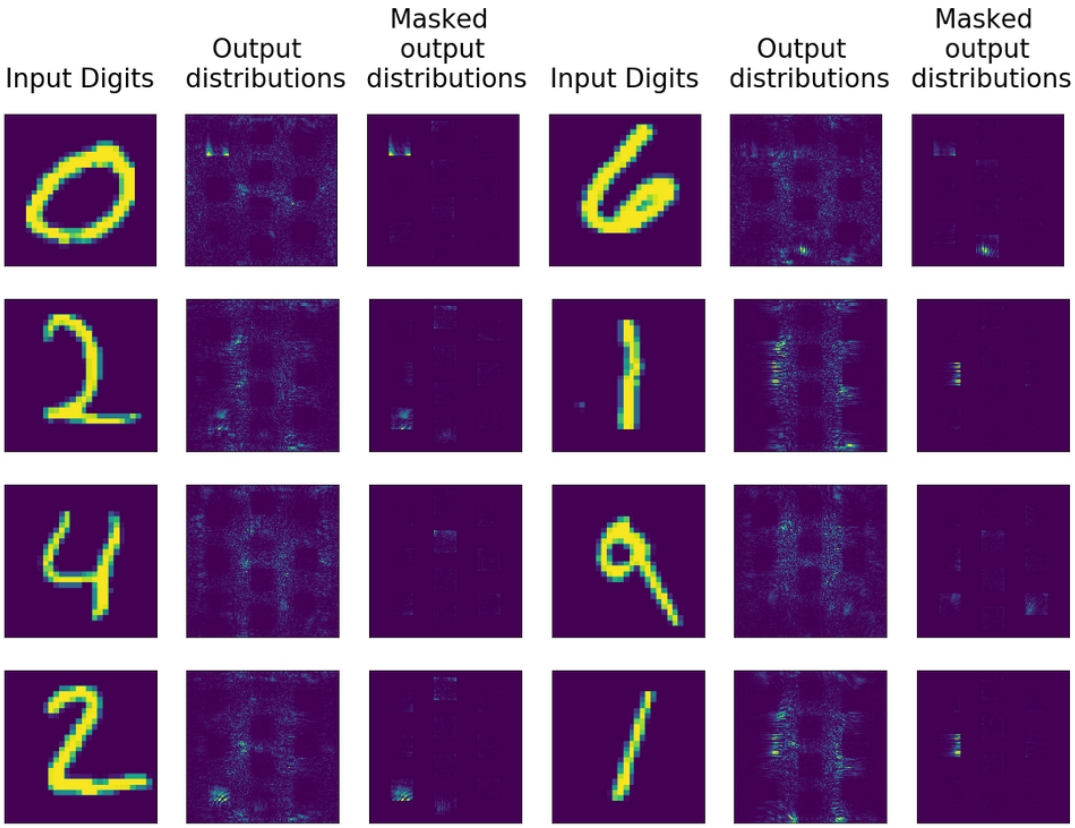


Figure 2.16: Data from test set, referred to a network with two layers with 200×200 pixels, $\delta x = 5um$, $d = 0.5cm$, detectors of 4×4 pixels.
Error E is evaluated through a not normalized output error. Training performed with 5 epochs and batches of $BS = 8$ gives a classification accuracy $\alpha_{tr} \approx 79.2\%$.
In each of the two semicolons, from left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied.

The confusion matrix and training accuracy as a function of the epoch number are represented respectively in Fig2.17 and Fig.2.18.

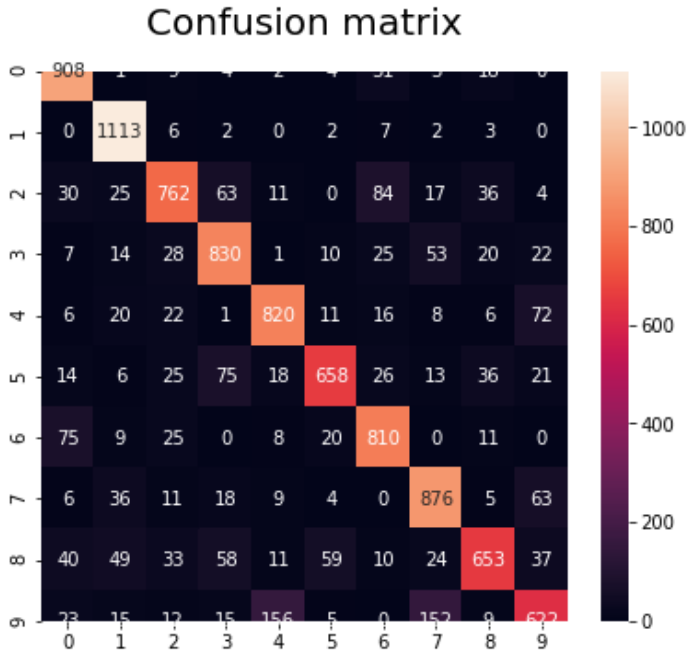


Figure 2.17: Confusion matrix; network with two layers of 200×200 pixels, $\delta_x = 5.0um$, $L = 1.0cm$, $d = 0.5cm$; training performed with 5 epochs, batch size=8

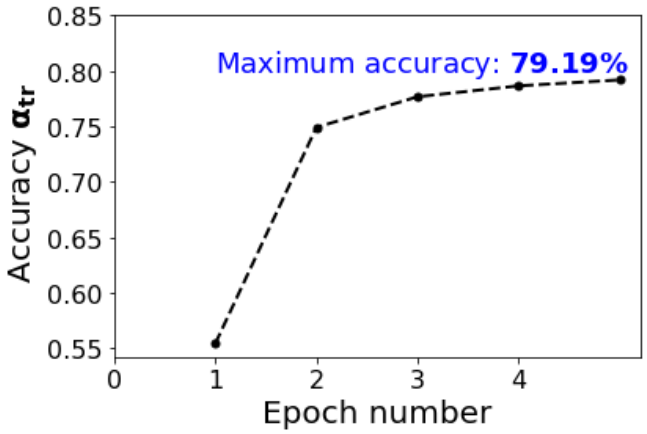


Figure 2.18: Classification accuracy α_{tr} as a function of the epoch number; network with two layers of 200×200 pixels, $\delta_x = 5.0um$, $L = 1.0cm$, $d = 0.5cm$

All in all, the above analyzed case turns out to be the one most satisfying, as far as possible, the conditions of full connectivity and exploitable geometrical features. The trained parameters, consisting in the phases of transmission coefficients of each diffractive

layer, give rise to two phase masks. The figure below represents their absolute values [5]:

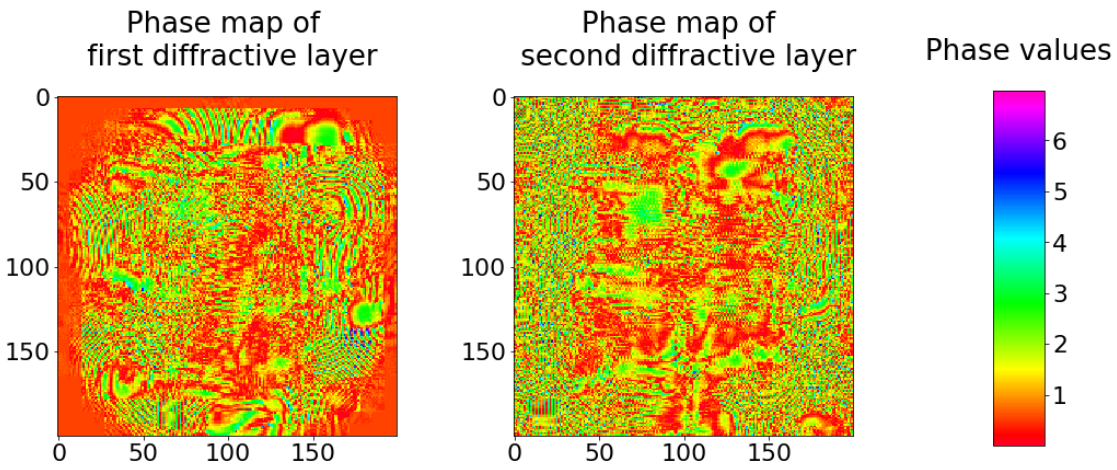


Figure 2.19: Phase mask of the first (on the left) and of the second (on the right) diffractive layers of side $L = 1.0cm$, constituted by 200×200 pixels and outdistanced of $d = 0.5cm$. Training has been performed with 5 epochs and $BS = 8$.

It is possible to observe from Fig.2.19 that the absolute values of the phases ϕ of complex transmission coefficients are included in a range extended from 0 to approximately 2π.
Each two phase mask must be converted into a height map, so as to fix the design of the network for its physical implementation.

[5]Modulus is taken for each phase value just in order to display clearer maps of the two diffractive layers

Chapter 3

Physical realization of the framework

The two matrixes containing the pixel by pixel height for each diffractive layer are analyzed Section 3.2. A brief description about two-photon lithography, the technique intended to be used to fabricate the planes, is instead provided in section 3.1.

3.1 Two-photon lithography (TPL)

Two-photon lithography is a well-established method based on two-photon polymerization (TPP) effect, whose most interesting advantage is to realize 3D complicated structures that can achieve higher resolution, until-$100nm$. So far, it has allowed to fabricate an extensive range of sophisticated nano-machines and photonic devices ([55],[56], [57], [58], [59], [60], [61]).
It is now briefly described the principle of photolithography, a key process employed in manufacturing of semiconductor devices and having a prominent role in the fabrication of micro-and nanostructures, typically accounting for about 30 percent of the cost of manufacturing.

3.1.1 Basics of photolithography

This technique uses UV light and basically transfers a pattern onto a substrate. As represented in Fig.3.1, at the beginning of a photolithographic process, the photoresist, a light-sensitive polymeric resin, is placed on the masking film to be etched, which is in turn deposited onto the substrate. Then, a mask of pattern, named as *photolithographic mask*, is positioned onto the structure, so that the photo-sensitive material can be modelled according to the desired shape, after exposure to electromagnetic radiation which modifies its chemical structure, hence its solubility. More specifically, a so called *positive resists* undergoes breaking down of its polymers, thus becoming more soluble in the developer,

the specific solution used to dissolve specific areas of the resin. In this way, the pattern remaining onto the wafer is an exact copy of the mask. On the contrary, negative photoresists become polymerized, hence more difficult to dissolve, under light exposure; in such case, the patterned resin is the "inverse" of the mask previously aligned onto it. Afterwards, etching is carried out giving a film of the desired shape and the remaining regions of the photoresist are finally eliminated.

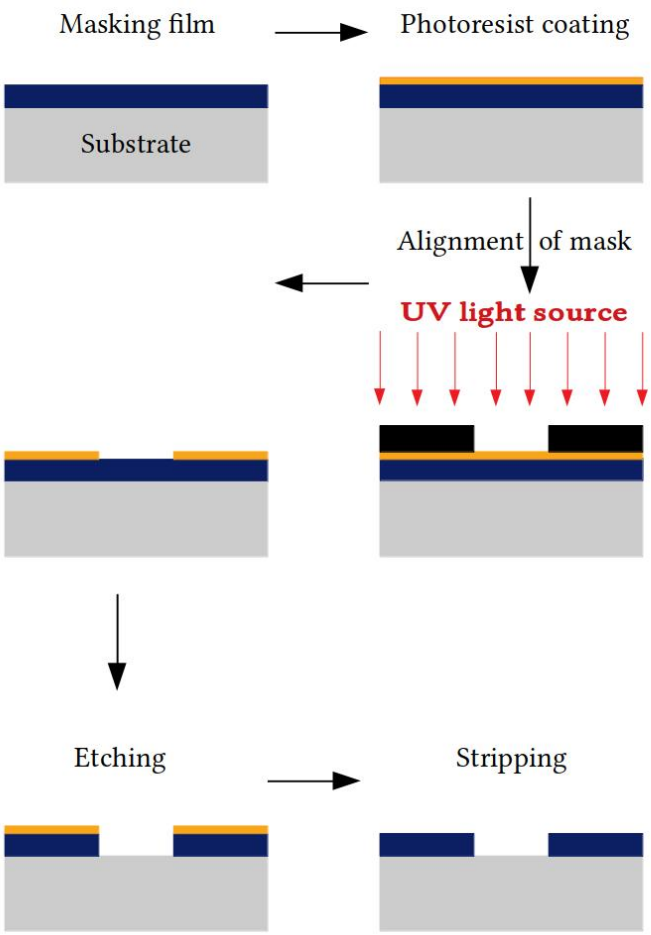


Figure 3.1: Photolithographic process

The pattern transferred to the photoresist through the lithographic mask is clearly two dimensional.
Another essentially planar lithographic technique is electron-beam lithography which

uses a focused beam of electrons, instead of light, to expose radiation-sensitive resists. It allows to obtain nm-scale features without the use of a lithographic mask, which is indeed constructed through such technique, used in fact for direct writing. Electron-beam lithography is a low-throughput technique, involving a serial scanning of the film to be patterned, differently from photolithography which represents a batch process enabling to expose in few seconds the entire surface of a substrate.
Three-dimensional structures can be obtained through planar lithographic techniques, like the ones just outlined, by reiterating several times, in sequence, the exposure step and the resist mask transfer step; nonetheless, this approach requires a considerable experimental effort. The technology of realizing three-dimensional structures by adding and exposing photo-sensitive resins layer by layer is classically named as "stereolithography"; UV laser stereolithography, as well as other 3D techniques including inject printing and the laser direct writing processes in common use ([62],[63]), can generate structures with a resolution of only few micrometers. Due to this, 3D lithography based on multi-photon polymerization is the currently used technique to obtain nanometric three-dimensional objects.
Conversely to the above mentioned techniques, TPL is intrinsically a 3D structuring process, since polymerization of the exposed resist occurs only in an extremely small area of the radiation-sensitive material, where the accumulated energy in the focus spot of an intense femto-second laser beam reaches the polymerization threshold. Defined as a mechanism for the fabrication of three-dimensional objects in 1997, when Maruo et al. ([64]) fabricated a spiral structure with diameter of about $7\mu m$, TPL combines two-photon absorption (TPA), whose theory was first developed in 1931 by Maria Goppert-Mayer in her doctoral thesis ([65]), with polymerization.

3.1.2 Two-photon absorption (TPA)

TPA is the simplest variant of multi-photon absorption. a process where more than one photon are absorbed simultaneously and excite an atom or an ion to a higher energetic state, with the energy increase equal to the sum of the photon energies. In multi-photon absorption processes of order n (i.e. involving absorption of n photons), the absorption rate is proportional to the n^{th} power of the optical intensity, therefore this is a nonlinear process, differently from one-photon absorption (OPA), where it is linear with the optical intensity.
This can be seen by considering that, in general photon absorption, from the optical field to the molecules of the medium an energy transfer occurs, which is quantified by the imaginary part of the susceptibility. In particular, the absolute value of polarization $\vec{P}$ of the medium is given by:

$$P = \chi^{(1)}E^1 + \chi^{(2)}E^2 + \chi^{(3)}E^3 + \chi^{(4)}E^4... \tag{3.1}$$

where $\vec{E}$ is the electric field of light interacting with the material and $\chi^{(1)}$, $\chi^{(2)}$, $\chi^{(3)}$, $\chi^{(4)}$ denote tensors representing linear, second-order third-order and forth-order optical

susceptibilities. Since in resonant processes there is no contribution from the even-order susceptibilities, like $\chi^{(2)}$ and $\chi^{(4)}$, the nonlinear absorption is described by the imaginary parts of $\chi^{(3)}$ and $\chi^{(5)}$, that affects respectively two-photon and three-photon absorption. Being the light-matter energy change per unit time and unit volume expressed as:

$$\frac{dW}{dt} = \left\langle \vec{E} \cdot \vec{P} \right\rangle \tag{3.2}$$

where square brackets stand for time average, it is obtained that for TPA the energy absorption rate is:

$$\frac{dW}{dt} \propto I^2 \, \Im(\chi^{(3)}) \tag{3.3}$$

with $\Im()$ denoting the "imaginary part". Thus, being the two-photon absorption rate quadratically dependent on light intensity I, it is nearly negligible for low or intermediate intensities, but becomes instead relevant for very high optical intensities, as happens with focused laser pulses. This explains why experimental results of TPA were not observed until 1961, by Kaiser and Garrett ([66]), relatively late with respect to observation in 1928 of Raman scattering [1], whose implementation needs lower intensities, since involving only one photon's absorption.
Two-photon absorption can be degenerate or nondegenerate, depending on whether the energy of the two photons is respectively the same or different. A representation of degenerate two-photon absorption process in comparison with OPA is provided in Fig.3.2:

[1] Two-photon process where one photon is absorbed and another is emitted essentially simultaneously; it consists of inelastic scattering of photons by matter and the difference between incident and scattered photons corresponds to vibrational energy levels of the molecule involved.

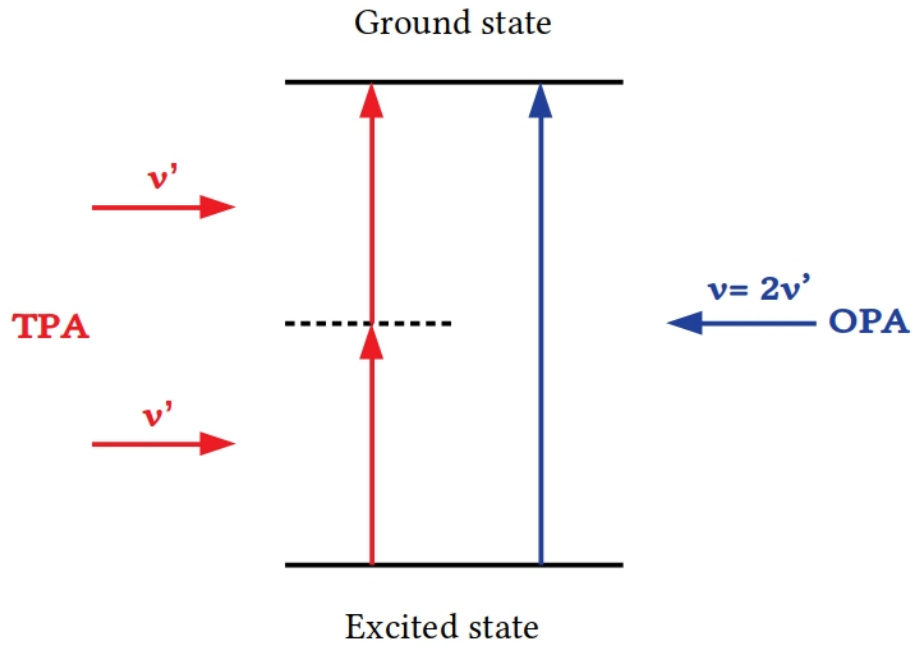


Figure 3.2: Scheme of degenerate two-photon absorption (TPA) process with $\nu' = \frac{\nu}{2}$ indicating photon frequency in the two-photon excitation beam (left) and of one-photon absorption (OPA) at frequency ν (right)

In TPA processes, electron excitations can occur stepwise or simultaneously. In the first case, a real intermediate state allows the further pumping of an already excited population to a higher level, as in two sequential single-photon absorptions, and coherence of the incident light is not required. In the latter, two photons are absorbed simultaneously by an electron, which acquires an energy exceeding the energy gap in one excitation event: a virtual state can be imagined to be formed when the first photon is absorbed. It remains for short time with respect to the intermediate energy level of simulataneous TPA and two-photon absorption occur if the second photon arrives before the collapse of this virtual state. Stepwise and simulataneous TPA are displayed in the figure below:

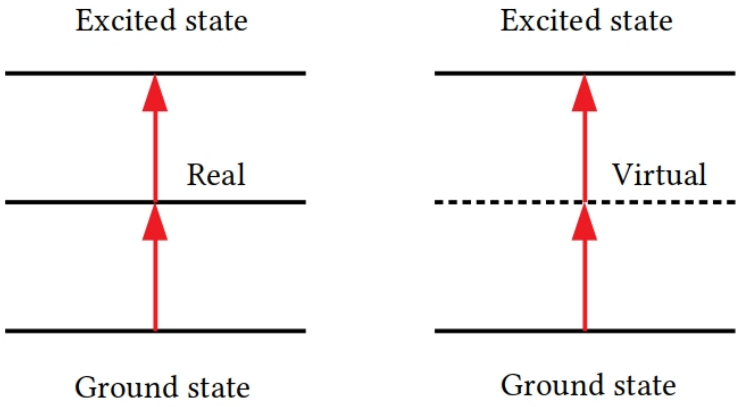


Figure 3.3: Scheme of two-photon absorption processes: stepwise on the left, simultaneous on the right

3.1.3 Photopolymerization

As previously mentioned, TPA is exploited in two-photon lithography to induce photopolymerization. Molecules more sensitive to light, including *photoinitiators* and *photosensitizers*, are generally added in photopolymers to increase the beginning productivity of monomers. Upon absorption of photons, photoinitiators, denoted as "I", form active species, like radicals ($R\cdot$), that can attack monomers or oligomers. This is the so called process of photoinitiation, described in the case of TPA as:

$$\mathrm{I} \xrightarrow{2h\nu'} \mathrm{I}^* \longrightarrow \mathrm{R}\cdot \tag{3.4}$$

where I^* represents an intermediate state of the photoinitiator after absorbing two photons of frequency ν'.
By considering also photosensitizers (S), namely molecules absorboing light and thransferring it to photoinitiators, the *photinitiationation* step is as described as:

$$\mathrm{S} \xrightarrow{2h\nu'} \mathrm{S}^* \ldots \xrightarrow{\mathrm{I}} \mathrm{I}^* \longrightarrow \mathrm{R}\cdot \tag{3.5}$$

The subsequent step is the *chain propagation*, in which monomer radicals, generated by the reaction of monomers or oligomers (M) with photoproduced radicals, combine with new monomers, and so forth; this process is expressed as:

$$\mathrm{R}\cdot + \mathrm{M} \longrightarrow \mathrm{R}\cdot\mathrm{M} \xrightarrow{\mathrm{M}} \mathrm{RMM}\ldots \longrightarrow \mathrm{R}\,\mathrm{M}_n \tag{3.6}$$

The so produced chain reaction is stopped by the meeting of two radicals, that establishes the step of *termination*, is manifested in one of the following ways:

$$\mathrm{RM}_n\cdot + \mathrm{RM}_m\cdot \longrightarrow \mathrm{RM}_{m+n}\mathrm{R} \tag{3.7}$$

$$\mathrm{RM}_n \cdot + \mathrm{RM}_m \cdot \longrightarrow \mathrm{RM}_m + \mathrm{RM}_n \tag{3.8}$$

3.1.4 Advantages of TPA for 3D lithography

TPA allows to achieve a striking polymerization control, differently from one-photon absorption. In the latter process it is used a conventional moderate-intensity source and excitation considerably weakens before the beam reaches the focal point, resulting in polymerization throughout the beam path in the sample. Instead, if a laser beam with wavelength so small as to induce TPA is used, only the molecules in close proximity to the beam's focus are excited, since, as previously described through Eq. 3.3, the energy absorption rate depends on the square of light intensity and, moreover, due to the fact that the section area of light beam increases with the distance from the focus, the light intensity approximately decreases quadratically with the distance z from the focal plane [2] along the propagation direction. As a result, the two-photon absorption rate diminishes with the forth power of z and the excitation of the material is maximal at the focus point and effectually falls off on both sides of the focal plane.
Since no attenuation of light in the photopolymer occurs until the beam reaches the focus, laser beam penetration is greatly enhanced with respect to polymerization by one-photon absorption, where the molecules excitation mitigates dramatically before the beam reaches the focal point, resulting in chemical reactions occurring only on top layer of the resin. In fact, as previously pointed out, OPA with UV radiation allows to fabricate 3D structures only using a layer-by-layer stereolithographic approach; moreover, the use of a mask is required when realizing 2D patterns through UV lasers and direct writing is possible only by means of electron-beam.
The great advantage provided by TPA is that the location of an excitation volume in the polymer can be controlled with high precision and resolution, allowing for very small volume (nearly "single point") polymerization and consequent extraordinarily narrow voxel (3D pixel) creation in the process of TPL.
Despite the TPA efficiency as wells as the thresholds of polymerization are determined by the nature of the particular photoresist, it is generally observed that resins intended to polymerize at UV or visible wavelengths, λ, can be polymerized at 2λ under the condition that the photon flux density provided by the radiation is high enough to initiate two-photon absorption. In particular, since photosensitive materials are usually transparent in the infrared range λ_{IR} and highly absorptive in the UV range (λ_{UV}), they can be polymerized by irradiation with the infra-red light of approximately double wavelength ($\lambda_{IR} = 2\lambda_{UV}$) by means of two-photon absorption. TPP with infrared laser pulses can thus be induced in extremely small volume, "voxels", of the material by focused near-IR femtosecond laser pulses. Any desired three-dimensional polymeric pattern is fabricated by direct "writing" into the volume of the photoresist. A simplified depiction of the difference

[2] The plane perpendicular to the light beam's optical axes, passing through the focal point, which is in turn the point where light rays converge.

between polymerization activated by one-photon and two-photon absorption is given in the figure below:

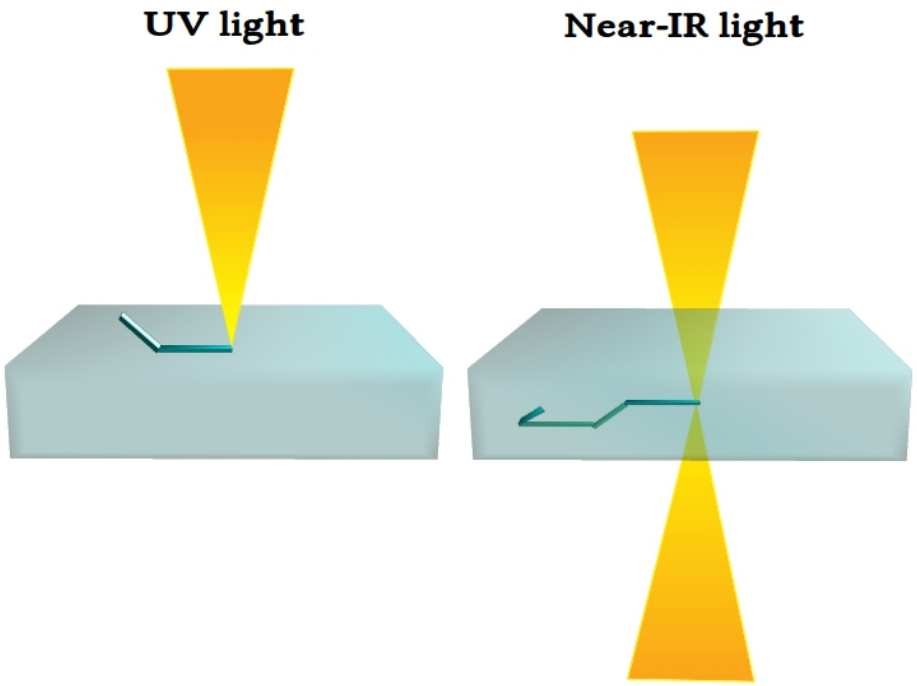


Figure 3.4: Simplified illustration of the difference between one-photon (on the left) and two-photons (on the right) activated polymerization processes, with darker blue "volumetric" traces representing the polymerized material.

3.2 Thickness maps of diffractive "layers"

The phase masks of each diffractive layer, obtained in Paragraph 2.3.4, are now converted to height maps through Eq.2.2, here again reported:

$$h = \frac{\lambda\phi}{2\pi\Delta n} \tag{3.9}$$

which gives the thickness h of each pixel whose trained parameter is represented by the phase ϕ.
As previously mentioned, Δn represents the difference between the refractive index of the resin which may be patterned by means of TPL and air ($n_{air} = 1$). With regards to the refractive index n_{resist} of photoresists typically used with femtosecond 3D direct

laser writers, it approximately takes value $n_{resist} \approx 1.5$, as results from measurements reported in the literature.
In particular, in the work performed by Gissibil et al. ([67]), the refractive indexes for five photoresists, namely IP-S, IP-Dip, IP-L, IP-G and OrmoComp, are obtained by measuring the critical angle of the total internal reflection for the used materials at different wavelengths. The results are reported in Fig.3.5:

Refractive indices	IP-S	IP-L	IP-G	OrmoComp	IP-Dip
$n_{450\ nm}$	1.5189	1.5272	1.5269	1.5315	1.5660
$n_{532\ nm}$	1.5103	1.5178	1.5187	1.5241	1.5534
$n_{588\ nm}$	1.5067	1.5140	1.5150	1.5207	1.5485
$n_{650\ nm}$	1.5039	1.5111	1.5120	1.5179	1.5446
$n_{780\ nm}$	1.4999	1.5074	1.5085	1.5140	1.5390
$n_{850\ nm}$	1.4985	1.5062	1.5072	1.5125	1.5367

Figure 3.5: Table extracted from [67], reporting the refractive index measurement of the photoresists Nanoscribe IP-Dip, micro resist OrmoComp, Nanoscribe IP-G, Nanoscribe IP-L, and Nanoscribe IP-S. The refractive index values are obtained by measuring the critical angle of the total internal reflection for the used photoresists at different wavelengths

Determined the value of $\Delta n = 0.5$, suitable for any material that may be used in the specific fabrication process, it is necessary to take into account the thickness resolution achieved in two-photon lithography. Considered that, before contacting a specific institution, this cannot be precisely known a priori and that a degree of generality is good to be maintained for the sake of experimental robustness, two possible "extreme" values of thickness resolution Δh have been analyzed.
The figure below illustrates the thickness maps of the first (on the left) and second (on the right) diffractive layers – of side $L = 1.0cm$, constituted by 200×200 pixels and outdistanced of $d = 0.5cm$ – whose phase values have been shown in Fig.2.19, after training performed with parameters established under the considerations described in Paragraph 2.3.4. It is emphasized that, for the sake of clarity in the image, the absolute values of thickness are represented, like done with the phases in Fig.2.19, despite a modellation process would occur from any side of each three-dimensional "layer" during fabrication.

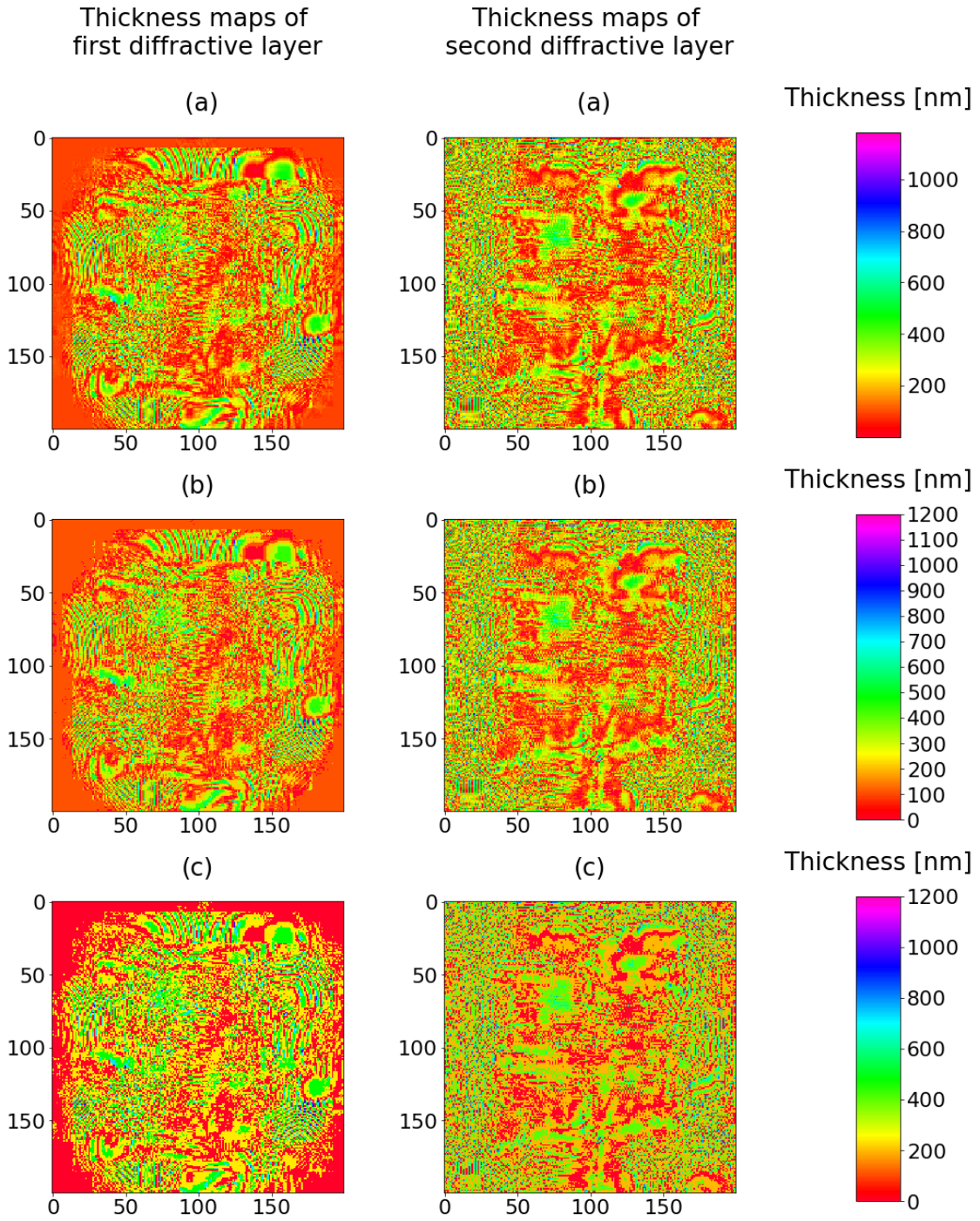


Figure 3.6: Thickness maps of the first (on the left) and of the second (on the right) diffractive layers of side $L = 1.0cm$, constituted by 200×200 pixels and outdistanced of $d = 0.5cm$. The network has been trained as described in Paragraph 2.3.4. Figures (a) represent the map directly obtained from the two phase values of Fig.2.19 without performing approximations provided by resolution, while (b) and (c) show approximated values which reflect a resolution in thickness of respectively $100nm$ and $200nm$.

Figures 3.6(a) show the two diffractive layers' distribution of thickness, where each pixel value, obtained through Eq.2.2 and expressed in nm, is not approximated: this "ideal" map clearly cannot reproduce a real fabricated object, since 2PL resolutions concern at least some tens of nanometers; nonetheless it is useful for comparison with images below.
Figures 3.6(b) display approximated values h reflecting a fabrication resolution $\Delta h = 100nm$: in particular, the following mapping has been applied to each positive value $h > 0$:

- $h \in [0, 50) \implies h = 0nm$
- $h \in [50, 150) \implies h = 100nm$
- $h \in [150, 250) \implies h = 200nm$
- ... and so on and so forth, throughout the whole range of thicknesses which ranges approximately from $0nm$ to $1053nm$ in the first layer and from $0nm$ to $1184nm$ in the second one.

Clearly, only a small part of the range of colors displayed in bar (b) appears in the related maps. As an example, considering the first diffractive plane, some pixels at the corners of the square, like the one at position (0,0), take value $h \approx 84nm$, which, according to the color bar on top (a), is represented with tending to red-dark orange color. The same regions have instead thickness $h = 100nm$ in maps (b), resulting in a lighter orange shade.

Figures 3.6(c) display approximated values h reflecting a fabrication resolution $\Delta h = 200nm$, leading to the mapping reported below:

- $h \in [0, 100) \implies h = 0nm$
- $h \in [100, 300) \implies h = 200nm$
- $h \in [300, 500) \implies h = 400nm$
- ... and so on and so forth, throughout the whole range of thicknesses.

The pixels at position (0,0) are now dark red, which is indeed the color corresponding to the value $h = 0nm$ in bar (c).
As one can observe, figures 3.6(c) show less convoluted patterns and lower gradation of colors/thickness with respect to 3.6(b), clearly due to an increasing homogenisation in pixel values occurring from the top to the bottom of the figure.

Thickness resolutions $\Delta h = 100nm$ and $\Delta h = 200nm$ correspond to phase resolution $\Delta\phi = \frac{2\pi\Delta h\Delta n}{\lambda}$ approximated respectively as 0.6 and 1.2. By approximating the phase masks' values according to such resolutions $\Delta\phi$ and inserting the so obtained diffractive layers in the network with geometrical parameters presented in 2.3.4, results

analogous to the one of Fig.3.7 are obtained at the output layer, after simulating light propagation through the framework.
In this particular example, it is possible to observe that the net lighting of detector associated to digit "2", characterizing the output distribution shown in Fig.3.7(a), weakens, in comparison to the rest of the plane, in (b) and more relevantly in (c), where pixels' phase resolutions $\Delta\phi$ correspond respectively to thickness ones of $\Delta h = 100nm$ and $\Delta h = 200nm$. This is clearly expected due to the approximation, more rude as proceeding from the top to the bottom of the figure, of values of transmission coefficients which have been previously trained to perform the desired task of image recognition. Nonetheless, the correct detector still displays an appreciable light intensity, as highlighted by the output distribution resulting upon application of the proper detector mask, which in fact obscures the other "confusing" illumination patterns arising in the (unmasked) output layers (b) and (c).

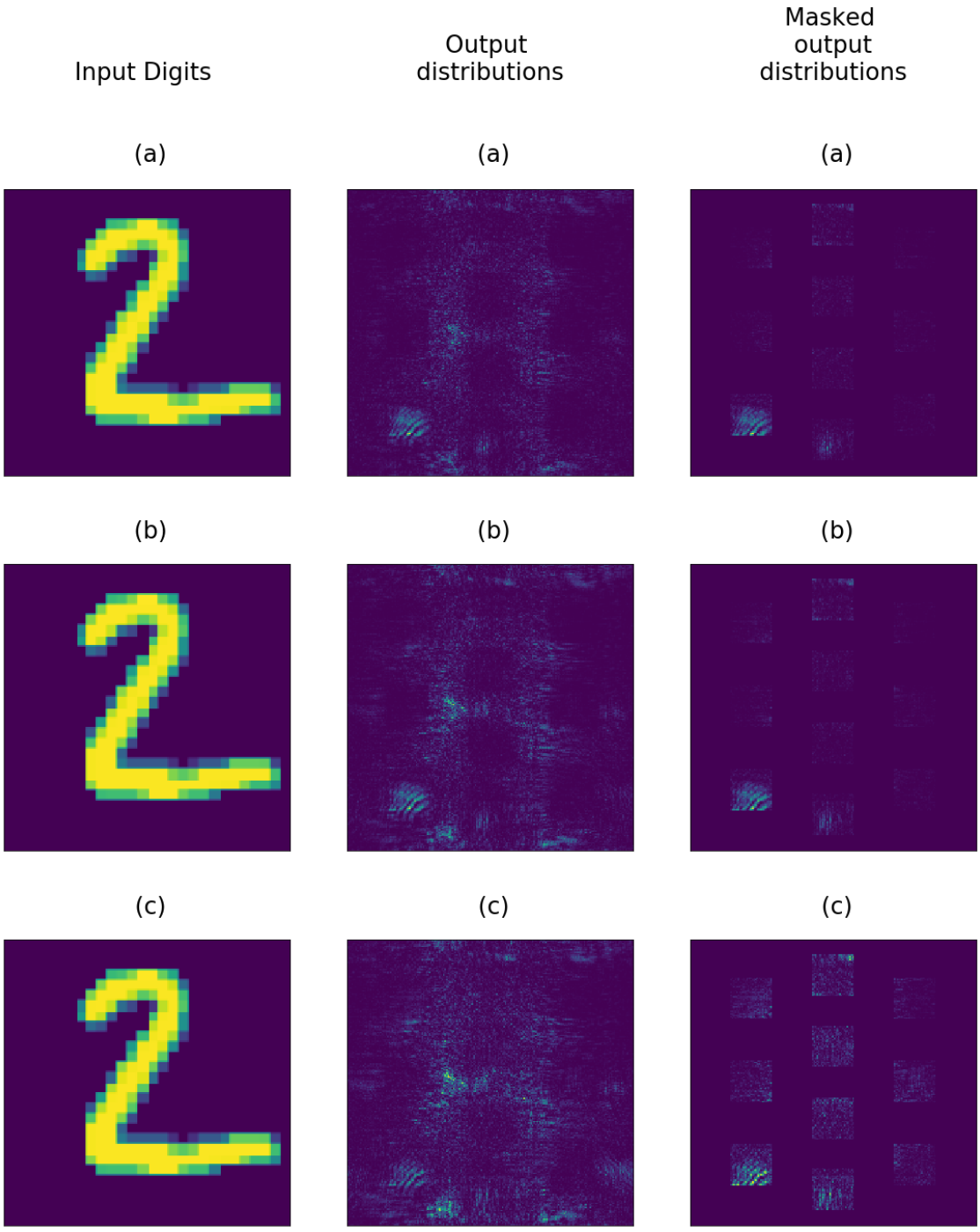


Figure 3.7: From left to right: input digit, intensity distribution incident on the output plane, intensity distribution on the output plane after a detectors mask has been applied, in a network with diffractive layers with 200×200 pixels, $\delta x = 5um$, $d = 0.5cm$, detectors of 4×4 pixels. Results (a), (b), (c) are obtained by numerically simulating the framework to be physically implemented, where light diffraction is performed by the two layers with thickness masks reported respectively in 3.6 (a), (b), (c).

According to numerical analysis, a lithographic resolution of $200nm$ would still be expected to lead the physical implemented network to perform digit classification with a sufficient degree of accuracy.

Conclusions

It has been trained a neural network with layers collaboratively performing digit classification, through diffracting an electromagnetic radiation with wavelength of $532\ nm$, firstly projected onto the image to be recognized and then free-space propagated between any two planes, from the input to the output one.
Two layers subject to learning are numerically subdivided into small finite elements: each of these implements Huygens-Fresnel principle, acting as source of a secondary wave when reached by luminous disturbance, and further introduces a *trainable* phase shift, identifying the transmission coefficient of a single pixel. The comprehensive fine-tune of such pixel-by-pixel functions, numerically identified by matrices, exactly determines the automatic learning of the system. This latter is in fact trained to modulate light so as to focus it onto one specific detecting region of the output plane, depending on the nature of the input image.
The resulting framework, since composed of two layers purely implementing linear optical functions, is all-optical and "deep". Despite composing more linear functions is equivalent to a unique linear operation, corresponding to the collapse of the multilayered structure into a single plane (hence the quotes around "deep"), integrating multiple planes proves to increase classification accuracy, due to involvement of more coefficients specifically predisposed to achieve the desired task. Our diffractive "deep" neural network (D^2NN) shows to reach an accuracy of approximately 81.3% when two diffractive layers of relatively few pixels, 28×28, are used. This represents an appreciably satisfying performance, considering that deep learning models classically integrate much more layers containing, on average, a number of pixels two orders of magnitude greater.

Such preliminary result confirms the reliability of our network for further exploration, which simultaneously concerns optimization of performances and setting of geometrical parameters suitable for its "real" implementation.
The system is, in fact, devised for being physically exploited in a laboratory: by converting the *trained* phases of transmission coefficients into a thickness map for both the diffractive layers, these latter, once 3D-printed through two-photon lithography, can be integrated in a specific experimental setup and collaboratively perform the precise classification of digits they have "virtually" been trained for. The performance of the network can thus be experimentally tested by illuminating an input image and observing its projection onto the

output layer: here, maximal brightness should be visible in correspondence of a specific detecting region, intrinsically associated (during the training phase) to the class of the digit to be recognized.

The network has thus been entirely designed in view of its experimental application, involving interpenetration in the present work between strategies, recurrent in deep learning's models, to improve the system performance and studies focusing on the robustness of relevant physical parameters of the framework.
In particular, the system has been setted so as to maximally satisfy the conditions of full-connectivity between layers (more suitable for numerical implementation), possible experimental alignment of parallel layers and their comfortable handling. Considering, according to Huygens principle, diffraction performed by each pixel, the first requirement implies a small enough ratio between dimension of each layer and its distance from the adjacent one; the second sufficiently short spacing between two consecutive planes and the third their large enough dimensions. Furthermore, pixels size is simultaneously desired to be abundantly small, so as to maximize diffraction-based connectivity between layers, and large enough, in order to minimize, for the sake of time computation, the number of neurons necessary to form an appreciably extended layer. Progressive analysis, developed on the basis of numerical results obtained in a sort of "trial" and "error" approach, have included exploration of various sizes of layers, pixels and detecting regions, as well as number of pixels and distances between consecutive planes.
The final result is represented by two phase masks associated to two diffractive squared layers of size $1.0\ cm$, outdistanced of $0.5\ cm$ and containing 200×200 pixels of area $5 \times 5\ um^2$. Classification learning is satisfactorily quantified by a training accuracy of 79.2%, reflecting into the net lighting up of a specific region of the output plane, which definitely denotes possibility of experimentally testing the system's performance.
Subsequent examinations of the related thickness maps reveal that even a (thickness) resolution of $200\ nm$, in two-photon lithographic process, should allow the physically implemented network to perform classification with sufficient degree of accuracy.

Our study constitutes not only the design of a powerful device performing digit classification at the speed of light, but also the base for further developments, involving, for instance, realization of optical deep neural networks achieving considerably sophisticated tasks through implementation of non-linear optical functions, by means of various materials, such as crystals, organic films, semiconductor materials. These, undoubtedly more challenging to be numerically modelled as well as experimentally exploited, with respect to the photoresist characterizing our potential layers, and to be numerically modelled, might lay the foundation for promising future studies.

Appendix: diffraction basics

.1 The origin of the phenomenon

According to the definition provided by Arnold Sommerfeld, "diffraction" refers to any phenomenon concerning deviation of a ray of light not caused by reflection or refraction. As is well known, diffraction can be classically observed, besides in several phenomena of daily life, when it is considered a spherical wave incident on a diaphragm presenting an opening, which is small enough but bigger than the size of the light source, indeed represented as a point source in the approximation of spherical wave. If a white screen is posed beyond the diaphragm, some fringes characterized by maxima and minima of light intensity are observed close to the bright spot's boundaries.
Diffraction therefore occurs when an obstacle or a slit is posed along the propagation path of a wave. As shown in the figure below, simulating the diffraction of a planewave from a slit, at the boundaries of the latter a discontinuity of the wavefront occurs: the wave bends at the extremes of the slit and a continuous perturbation is formed.

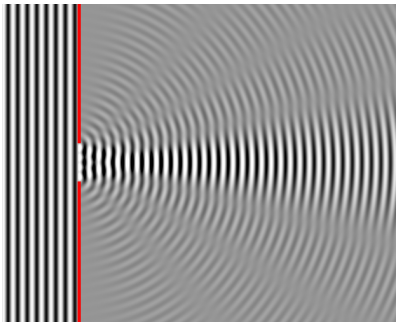


Figure 8: Simulation of diffraction of a planewave from a slit; image taken from [68]

The Huygens-Fresnel principle, expressed through Eq.1.22 and stating that any infinitesimal element of a wavefront can be formally regarded as a secondary source of spherical waves in phase with the primary wave and with amplitude proportional to the one of primary wave, constitutes the key for interpreting diffraction. According to it, when the wave meets an obstacle, its configuration at any point P' of the space beyond

the obstacle needs to be calculated by summing up the contributions of elementary spherical waves emitted by each point of the wavefront in presence of the obstacle. Differently from the case of free-space propagation where all the non radial contributes, coming from a wavefront at position P cancel out with each other in P' due to destructive interference, in presence of an obstacle or a slit it is not known a priori the wavefront's configuration in a given point beyond it. The wave perturbation at any point is thus the sum of secondary waves generated from the wavefront in a previous position of its propagation path: the observed diffraction fringes, consisting in maxima and minima of light intensity, result from the range of values that the total amplitude, determined by relative phases and amplitudes of the contributes, can span.

The foundation of the theory mathematically describing and explaining such phenomenon was laid in 1678 by Huygens, who manifested his intuition about the identification of each point of a wavefront as a new source of a secondary spherical perturbation. In 1818, Fresnel unified such idea with interference concepts in the meantime provided by Young, by calculating with relatively high precision the distribution of light in observed diffraction fringes, basing on some assumptions about the amplitudes and phases of the secondary waves. Seventy-four years later, Kirchoff built a more solid mathematical formulation for diffraction theory. The wave diffracted by an aperture is rigorously calculated from the Kirchhoff diffraction equation, whose parameters had to be arbitrarily assigned in the derivation of the Huygens–Fresnel equation. Analytical solution of Kirchoff equation are not possible for several configurations: Fresnel and Fraunofer diffractions are useful to simplify calculations, while nevertheless exhaustively illustrating the core of diffraction theory, suiting for the propagation of waves respectively in the near and far fields.

We now consider the simplifications led by Fresnel and Fraunofer approximations to the diffraction equation 1.22, reported below:

$$\begin{aligned} U(x,y;z) &= \iint_{-\infty}^{+\infty} U_0(x',y';0)\ h(x-x',y-y',z)\ dx'\,dy' = \\ &= -\frac{j}{\lambda}\iint_{-\infty}^{+\infty} U_0(x',y';0)\ \frac{e^{jkr}}{r}\ \frac{z}{r}\ dx'\,dy' \end{aligned} \tag{10}$$

that is used to calculate the field $U(x,y;z)$, at plane (x,y) placed in position z along the optical axes, as a (continue) sum of secondary waves generated by the points of a diffractive aperture, defined over the plane (x',y') located at $z=0$; such geometry is represented in Fig.9.

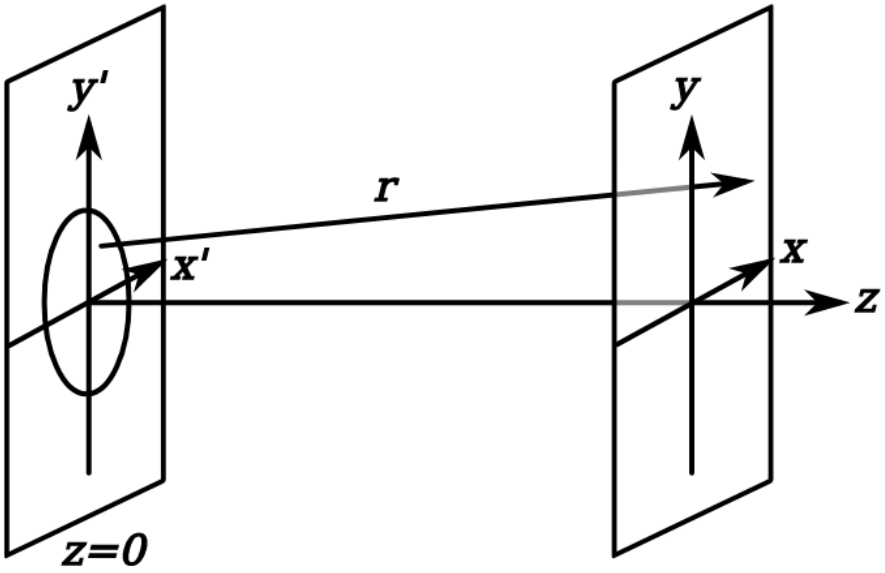


Figure 9: Diffraction geometry: an aperture, acting as diffracting object on input plane, and image plane are represented, within a Cartesian coordinate system.

The distance between two given points $P' = (x', y'; z = 0)$ and $P = (x, y; z)$ is provided by

$$r = \sqrt{(x-x')^2 + (y-y')^2 + z^2} = z\sqrt{1 + \frac{(x-x')^2}{z^2} + \frac{(y-y')^2}{z^2}}$$

By exploiting the Fresnel approximation

$$x - x', y - y' \ll z$$

, meaning that the lateral extension of the region explored by light propagating between the two planes is smaller than its length, r can be approximated by the binomial expansion as:

$$r \approx z\left(1 + \frac{1}{2}\frac{(x-x')^2}{z^2} + \frac{(y-y')^2}{z^2}\right)$$

which basically leads to approximating a spherical wavefront into a parabolical wavefront. In this way, it is possible to write:

$$U(x, y; z) \approx \frac{e^{jkz}}{j\lambda z} \iint_{x',y'} U_0(x', y'; 0)\ e^{j\frac{k}{2z}\left[(x-x')^2 + (y-y')^2\right]}\ dx'\, dy' \tag{11}$$

Eq.11 can be rewritten as:

$$U(x, y; z) \approx \frac{e^{jkz}}{j\lambda z}\, e^{\left[\frac{jk}{2z}(x^2+y^2)\right]} \iint_{x',y'} U_0(x', y'; 0)\ e^{\left[\frac{jk}{2z}(x'^2+y'^2)\right]}\, e^{\left[\frac{-j2\pi}{\lambda z}(xx'+yy')\right]}\ dx'\, dy' \tag{12}$$

which, by considering the spatial frequencies $\frac{x}{\lambda z} = f_x$, $\frac{y}{\lambda z} = f_y$, is in turn expressed as

$$U(x,y;z) \propto \mathcal{F}\left\{U(x,y,0)\, e^{j\frac{k}{2z}(x'^2+y'^2)}\right\} \tag{13}$$

Considering, ulteriorly, Fraunhofer approximation, given by:

$$x'^2 + y'^2 \ll z$$

and expressing that the diffraction pattern is viewed at a long distance from the diffracting object or also at the focal plane of an imaging lens, from Eq.12 it is obtained:

$$U(x,y;z) \propto \iint_{x',y'} U_0(x',y';0)\; e^{\left[\frac{-j2\pi}{\lambda z}(xx'+yy')\right]}\; dx'\, dy' \tag{14}$$

thus

$$U(x,y;z) \propto \mathcal{F}\{U(x,y,0)\}$$

which fundamentally expresses that, by illuminating an object (for instance, a circular aperture) with a monochromatic wave and observing very far away from the object the pattern distribution, the latter is simply the Fourier transform of the object (the Airy pattern, in case of a circular hole).

.2 Diffraction from a single slit of infinite length

The most elementary configuration that can be considered as a source of diffraction is the single slit, here analyzed since at the base of notions reported in the main text.
A monochromatic wave

$$U(x,y;0) = u\, e^{jkct} = a\, e^{j\frac{2\pi ct}{\lambda}}$$

, with u, λ, c, t denoting respectively magnitude of the wave disturbance, wavelength, velocity of light and time, is incident onto a diaphragm containing a small opening of width a and length $h \gg a$, where the latter extends along the direction orthogonal to the plane of observation. The waves propagating along the direction, forming an angle θ with respect the perpendiculars of the diaphragm, sum up giving rise to a wave that can be visualized by means of a lens onto the screen, posed at its focal plane and at a distance D from the diaphragm. Therefore, Fraunhofer approximation is allowed.
With reference to Fig.9, assuming the center of the slit located at $x = 0$, the expression 14, for each x value, gives the following expression for the amplitude field distribution on a screen at a distance D, along z direction, from the slit:

$$U(x,y;D) = u \int_{-\frac{a}{2}}^{+\frac{a}{2}} e^{\left[\frac{-j2\pi}{\lambda z}(yy')\right]}\; dy' \;= ua\ \mathrm{sinc}\left(\frac{\pi ua}{\lambda D}\right) \tag{15}$$

where the obtained sinc function is represented below:

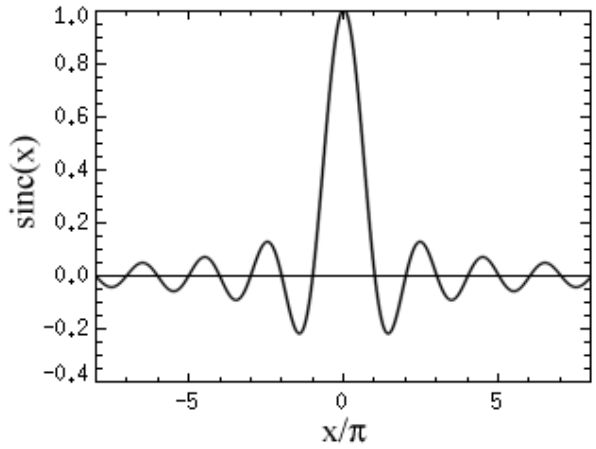


Figure 10: Sinc function, describing the amplitude of the field distribution determined by single slit diffraction

Clearly, the measurable intensity of the diffraction pattern will be given by the modulus square of $U(x, y; D)$.
Fig.11 displays the just described configuration:

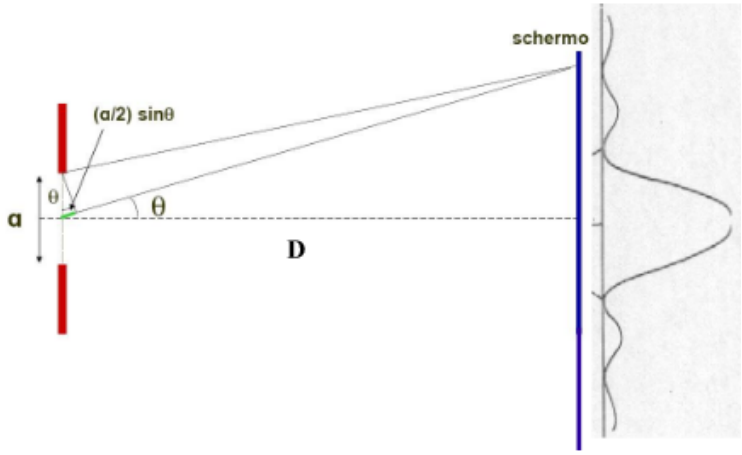


Figure 11: Single slit diffraction

$\frac{a}{2}sin(\theta)$ represents the phase displacement between the "secondary wave" generated by the upper extreme of the opening and the one coming from the shit's central point. The position of the first minimum of intensity in the diffraction pattern, formed onto the screen, is found by imposing destructive interference between the two rays in correspondence of such plane, namely:

$$\frac{a}{2}\, sin(\theta) = \frac{\lambda}{2} \tag{16}$$

which gives

$$a\ sin(\theta) = \lambda \tag{17}$$

which can, with good enough approximation, quantify the half-cone diffraction angle θ of Eq.2.19, determining the position of the bright central region of the intensity pattern generated on a given plane l, upon diffraction by one of the N^2 pixels of size δx contained in layer $l-1$.

References

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539.

[2] A. B. Nassif et al. "Speech Recognition Using Deep Neural Networks: A Systematic Review". In: *IEEE Access* 7 (2019), pp. 19143–19165. DOI: 10.1109/ACCESS.2019.2896880.

[3] Ali Nassif et al. "Speech Recognition Using Deep Neural Networks: A Systematic Review". In: *IEEE Access* PP (Feb. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2896880.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation. 2014. URL: http://arxiv.org/abs/1409.0473.

[5] Marta R. Costa-jussà et al. "Introduction to the special issue on deep learning approaches for machine translation". In: *Computer Speech and Language* 46 (2017), pp. 367 –373. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2017.03.001. URL: http://www.sciencedirect.com/science/article/pii/S0885230816303965.

[6] Wanli Ouyang et al. "DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[7] J. Han et al. "Advanced Deep-Learning Techniques for Salient and Category-Specific Object Detection: A Survey". In: *IEEE Signal Processing Magazine* 35.1 (2018), pp. 84–100. DOI: 10.1109/MSP.2017.2749125.

[8] Dumitru Erhan et al. "Scalable Object Detection using Deep Neural Networks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[9] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. "Deep Neural Networks for Object Detection". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 2553–2561. URL: http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf.

[10] James Zou et al. "A primer on deep learning in genomics". In: *Nature Genetics* 51 (Nov. 2018). DOI: 10.1038/s41588-018-0295-5.

[11] Yongjin Park and Manolis Kellis. "Deep learning for regulatory genomics". In: *Nature biotechnology* 33 (Aug. 2015), pp. 825–6. DOI: 10.1038/nbt.3313.

[12] Gökcen Eraslan et al. "Deep learning: new computational modelling techniques for genomics". In: *Nature Reviews Genetics* 20 (Apr. 2019), p. 1. DOI: 10.1038/s41576-019-0122-6.

[13] Erik Gawehn, Jan A. Hiss, and Gisbert Schneider. "Deep Learning in Drug Discovery". In: *Molecular Informatics* 35.1 (2016), pp. 3–14. DOI: 10.1002/minf.201501008. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/minf.201501008. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201501008.

[14] Hongming Chen et al. "The rise of deep learning in drug discovery". In: *Drug Discovery Today* 23.6 (2018), pp. 1241 –1250. ISSN: 1359-6446. DOI: https://doi.org/10.1016/j.drudis.2018.01.039. URL: http://www.sciencedirect.com/science/article/pii/S1359644617303598.

[15] Izhar Wallach, Michael Dzamba, and Abraham Heifets. "AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery". In: (Oct. 2015).

[16] Yann LeCun and Corinna Cortes. "MNIST handwritten digit database". In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

[17] J.W. Goodman. *Introduction to Fourier Optics*. McGraw-Hill Series in Electrical and Computer Engineering: Communications and Signal Processing. McGraw-Hill, 1996. ISBN: 9780070242548. URL: https://books.google.it/books?id=QllRAAAAMAAJ.

[18] Xing Lin et al. "All-optical machine learning using diffractive deep neural networks". In: *Science* 361.6406 (2018), pp. 1004–1008. DOI: 10.1126/science.aat8084.

[19] Xing Lin et al. "Supplementary Material for all-optical machine learning using diffractive deep neural networks". In: *Science* (2018). DOI: 10.1126/science.aat8084.

[20] Waseem Rawat and Zenghui Wang. "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review". In: *Neural Computation* 29 (June 2017), pp. 1–98. DOI: 10.1162/NECO_a_00990.

[21] V. V. Shakirov, K. P. Solovyeva, and W. L. Dunin-Barkowski. "Review of State-of-the-Art in Deep Learning Artificial Intelligence". In: *Optical Memory and Neural Networks* 27.2 (2018), pp. 65–80. ISSN: 1934-7898. DOI: 10.3103/S1060992X18020066. URL: https://doi.org/10.3103/S1060992X18020066.

[22] Nabil H. Farhat et al. "Optical implementation of the Hopfield model". In: *Appl. Opt.* 24.10 (1985), pp. 1469–1475. DOI: 10.1364/AO.24.001469. URL: http://ao.osa.org/abstract.cfm?URI=ao-24-10-1469.

[23] Demetri Psaltis and Nabil Farhat. "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback". In: *Opt. Lett.* 10.2 (1985), pp. 98–100. DOI: 10.1364/OL.10.000098. URL: http://ol.osa.org/abstract.cfm?URI=ol-10-2-98.

[24] Bahram Javidi, Jian Li, and Qing Tang. "Optical implementation of neural networks for face recognition by the use of nonlinear joint transform correlators". In: *Appl. Opt.* 34.20 (1995), pp. 3950–3962. DOI: 10.1364/AO.34.003950. URL: http://ao.osa.org/abstract.cfm?URI=ao-34-20-3950.

[25] Demetri Psaltis, David Brady, and Kelvin Wagner. "Adaptive optical networks using photorefractive crystals". In: *Appl. Opt.* 27.9 (1988), pp. 1752–1759. DOI: 10.1364/AO.27.001752. URL: http://ao.osa.org/abstract.cfm?URI=ao-27-9-1752.

[26] A.N. Tait et al. "Broadcast and Weight: An Integrated Network For Scalable Photonic Spike Processing". In: *Journal of Lightwave Technology* 32 (Nov. 2014). DOI: 10.1109/JLT.2014.2345652.

[27] Demetri Psaltis et al. "Holography in artificial neural networks". In: *Landmark Papers on Photorefractive Nonlinear Optics*, pp. 541–546. DOI: 10.1142/9789812832047_0076. eprint: https://www.worldscientific.com/doi/pdf/10.1142/9789812832047_0076. URL: https://www.worldscientific.com/doi/abs/10.1142/9789812832047_0076.

[28] A.N. Tait et al. "Neuromorphic photonic networks using silicon photonic weight banks". In: *Scientific Reports* 7 (Dec. 2017). DOI: 10.1038/s41598-017-07754-z.

[29] Y. Shen et al. "Deep learning with coherent nanophotonic circuits". In: *2017 IEEE Photonics Society Summer Topical Meeting Series (SUM)*. 2017, pp. 189–190. DOI: 10.1109/PHOSST.2017.8012714.

[30] David Rosenbluth et al. "A high performance photonic pulse processing device". In: *Opt. Express* 17.25 (2009), pp. 22767–22772. DOI: 10.1364/OE.17.022767. URL: http://www.opticsexpress.org/abstract.cfm?URI=oe-17-25-22767.

[31] Min Gu et al. "Optically Digitalized Holography: A Perspective for All-Optical Machine Learning". In: *Engineering* 5 (Apr. 2019). DOI: 10.1016/j.eng.2019.04.002.

[32] Durgesh Srivastava and L. Bhambhu. "Data classification using support vector machine". In: *Journal of Theoretical and Applied Information Technology* 12 (Feb. 2010), pp. 1–7.

[33] R. Maree et al. "Random subwindows for robust image classification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 34–40 vol. 1. DOI: 10.1109/CVPR.2005.287.

[34] Hoang Le et al. "Image Classification using Support Vector Machine and Artificial Neural Network". In: *International Journal of Information Technology and Computer Science* 4 (May 2012). DOI: 10.5815/ijitcs.2012.05.05.

[35] X.-Z Qi and Q. Wang. "An image classification approach based on sparse coding and multiple kernel learning". In: 40 (Apr. 2012), pp. 773–779. DOI: 10.3969/j.issn.0372-2112.2012.04.025.

[36] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. "Exploiting Similarities among Languages for Machine Translation." In: *CoRR* abs/1309.4168 (2013). URL: http://dblp.uni-trier.de/db/journals/corr/corr1309.html#MikolovLS13.

[37] Libin Shen, Anoop Sarkar, and Franz Josef Och. "Discriminative Reranking for Machine Translation". In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Boston, Massachusetts, USA: Association for Computational Linguistics, 2004, pp. 177–184.

[38] Daxiang Dong et al. "Multi-Task Learning for Multiple Language Translation". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1723–1732. DOI: 10.3115/v1/P15-1166.

[39] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. "Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 866–875. DOI: 10.18653/v1/N16-1101.

[40] V. N. T. Truong, C. Yang, and Q. Tran. "A translator for American sign language to text and speech". In: *2016 IEEE 5th Global Conference on Consumer Electronics*. 2016, pp. 1–2. DOI: 10.1109/GCCE.2016.7800427.

[41] Diane Litman. "Classifying Cue Phrases in Text and Speech Using Machine Learning". In: (Feb. 1999).

[42] Sam Scott and Stan Matwin. "Text Classification Using WordNet Hypernyms". In: *Usage of WordNet in Natural Language Processing Systems*. 1998.

[43] Thiago S. Guzella and Walmir M. Caminhas. "A review of machine learning approaches to Spam filtering". In: *Expert Systems with Applications* 36.7 (2009), pp. 10206 –10222. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2009.02.037. URL: http://www.sciencedirect.com/science/article/pii/S095741740900181X.

[44] David Heckerman et al. "A Bayesian Approach to Filtering Junk E-Mail". In: *AAAI Workshop on Learning for Text Categorization*. 1998. URL: https://www.microsoft.com/en-us/research/publication/a-bayesian-approach-to-filtering-junk-e-mail/.

[45] Ion Androutsopoulos et al. "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach". In: (Oct. 2000).

[46] Xavier Carreras and Jordi Salgado. "Boosting Trees for Anti-Spam Email Filtering". In: (Oct. 2001).

[47] William W. Cohen. "Learning Rules that Classify E-Mail". In: *Proceedings of the 1996 AAAI Spring Symposium on Machine Learning and Information Access*. 1996.

[48] W.Patrick Walters and Mark A Murcko. "Prediction of 'drug-likeness'". In: *Advanced Drug Delivery Reviews* 54.3 (2002). Computational Methods for the Prediction of ADME and Toxicity, pp. 255 –271. ISSN: 0169-409X. DOI: https://doi.org/10.1016/S0169-409X(02)00003-0. URL: http://www.sciencedirect.com/science/article/pii/S0169409X02000030.

[49] In: ().

[50] In: ().

[51] Maxwell W. Libbrecht and William Stafford Noble. "Machine learning applications in genetics and genomics". In: *Nature Reviews Genetics* 16 (2015), pp. 321–332.

[52] Daniel R. Schrider and Andrew D. Kern. "Supervised Machine Learning for Population Genetics: A New Paradigm". In: *Trends in Genetics* 34.4 (2018), pp. 301 –312. ISSN: 0168-9525. DOI: https://doi.org/10.1016/j.tig.2017.12.005. URL: http://www.sciencedirect.com/science/article/pii/S0168952517302251.

[53] In: ().

[54] Haiqing Wei et al. *Comment on All-optical machine learning using diffractive deep neural networks*. Sept. 2018.

[55] Andreas Ostendorf and Boris Chichkov. "Two-photon polymerization: A new approach to micromachining". In: *Photonics Spectra* 40 (Oct. 2006), pp. 72–80.

[56] Gheorghe Cojoc et al. "Optical micro-structures fabricated on top of optical fibers by means of two-photon photopolymerization". In: *Microelectronic Engineering* 87 (May 2010), pp. 876–879. DOI: 10.1016/j.mee.2009.12.046.

[57] "G. Cojoc et al. ""Optical micro-structures fabricated on top of optical fibers by means of two-photon photopolymerization"". In: *"Microelectronic Engineering"* "87"."5" ("2010"). "The 35th International Conference on Micro- and Nano-Engineering (MNE)", "876 –879". ISSN: "0167-9317". DOI: "https://doi.org/10.1016/j.mee.2009.12.046". URL: "http://www.sciencedirect.com/science/article/pii/S0167931709008892".

[58] Dong Wu et al. "Femtosecond laser rapid prototyping of nanoshells and suspending components towards microfluidic devices". In: *Lab on a chip* 9 (Sept. 2009), pp. 2391–4. DOI: 10.1039/b902159k.

[59] Michael Thiel and Martin Hermatschweiler. "Three-dimensional laser lithography". In: *Optik and Photonik* 6 (Dec. 2011). DOI: 10.1002/opph.201190386.

[60] Maria Farsari and Boris Chichkov. "Two-photon fabrication". In: *Nature Photonics* 3 (Aug. 2009), pp. 450–452. DOI: 10.1038/nphoton.2009.131.

[61] Attilio Marino et al. "Two-Photon Polymerization of Sub-micrometric Patterned Surfaces: Investigation of Cell-Substrate Interactions and Improved Differentiation of Neuron-like Cells". In: *ACS Applied Materials and Interfaces* 5.24 (2013), pp. 13012–13021. DOI: 10.1021/am403895k.

[62] In: ().

[63] Victor Korolkov, Ruslan Nasyrov, and Ruslan Shimansky. "Optimization for direct laser writing of continuous-relief diffractive optical elements". In: *Applied optics* 45 (Feb. 2006), pp. 53–62. DOI: 10.1364/AO.45.000053.

[64] Shoji Maruo, Osamu Nakamura, and Satoshi Kawata. "Three-dimensional microfabrication with two-photon-absorbed photopolymerization". In: *Opt. Lett.* 22.2 (1997), pp. 132–134. DOI: 10.1364/OL.22.000132.

[65] M. Göppert-Mayer. "Elementary processes with two quantum transitions". In: *Annalen der Physik* 18.7-8 (2009), pp. 466–479. DOI: 10.1002/andp.200910358.

[66] W. Kaiser and C. G. B. Garrett. In: *Phys. Rev. Lett.* 7 (6 1961), pp. 229–231. DOI: 10.1103/PhysRevLett.7.229. URL: https://link.aps.org/doi/10.1103/PhysRevLett.7.229.

[67] Timo Gissibl et al. "Refractive index measurements of photo-resists for three-dimensional direct laser writing". In: *Opt. Mater. Express* 7.7 (2017), pp. 2293–2298. DOI: 10.1364/OME.7.002293. URL: http://www.osapublishing.org/ome/abstract.cfm?URI=ome-7-7-2293.

[68] Wikipedia. URL: \url{https://it.wikipedia.org/wiki/Diffrazione}.