



École Polytechnique Fédérale de Lausanne

Generative programming
of multilayer optical processor

by Petr Šicho

Semester Project Report

Project Advisors

Niyazi Ulas Dinç
Mustafa Yildirim

Project Supervisor

Moser Christophe

July 30, 2026

Abstract

Optical neural networks (ONNs) have recently emerged as a promising energy-efficient alternative to conventional electronic neural networks. In this project, we investigate a multilayer optical processor in which the phase modulation parameters are not learned directly through backpropagation, but are instead generated by an implicit neural representation. In particular, we employ a SIREN (Sinusoidal Representation Network) to generate smooth scale and bias masks for each optical layer. This generative approach significantly reduces the number of learnable parameters while improving performance, especially for deeper optical networks.

Contents

1	Introduction and Motivation	4
1.1	Background on Optical Computing	4
1.2	State of the Art	5
2	Goals and approach	6
2.1	SIREN: Implicit Neural Representations	6
2.2	Architecture	7
2.2.1	Generator Network Design	7
2.3	Optimization and Training	8
2.3.1	Hyperparameter Sensitivity	8
2.4	Data Processing	9
2.4.1	Dataset Description	9
2.4.2	Data Augmentation and Normalization	9
3	Results	10
3.1	Baseline Performance	10
3.2	Our results	10
3.3	Conclusion	11
	Bibliography	13

Introduction and Motivation

1.1 Background on Optical Computing

Optical computing exploits properties of light, such as diffraction and interference, to perform computations in the physical domain. Unlike electronic processors, which rely on sequential operations and suffer from significant energy requirements, optical systems naturally enable parallelism with very low energy dissipation during signal propagation. These characteristics make optical hardware particularly attractive for accelerating machine learning workloads and reducing its energy cost and environmental footprint

Optical neural networks (ONNs) implement neural computation by encoding network weights into optical elements such as phase masks on spatial light modulators (SLM). As light propagates through these elements, linear transformations equivalent to matrix multiplications are performed at the speed of light. While nonlinearities and programmability typically require electronic support, by repeating the data on each of the optical layer, like in Figure 1.1, we can implement a polynomial transformation of the input.

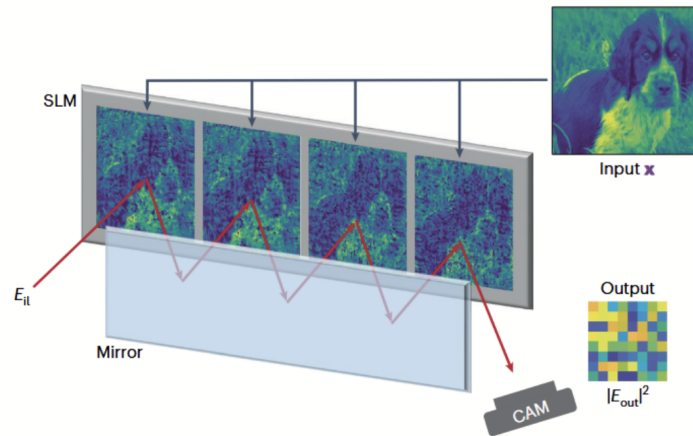


Figure 1.1: Conceptual overview of a multilayer optical neural network with four phase-modulating layers. Image taken from [2]

1.2 State of the Art

In this optical neural network architecture, multiple spatial light modulator (SLM) layers sequentially display a linearly-transformed variant of the input X . For each layer, we therefore need to choose two masks for scaling S and bias B to define the phase mask $P = S \cdot X + B$, where $\cdot$ represents element-wise multiplication. This phase mask is then displayed on the SLM and applied to the optical field, followed by free-space propagation. The more of these layers we have, the more complex transformations can we encode in the network. After the final optical layer, the output field intensity is spatially averaged and passed to a linear classifier that assigns a label.

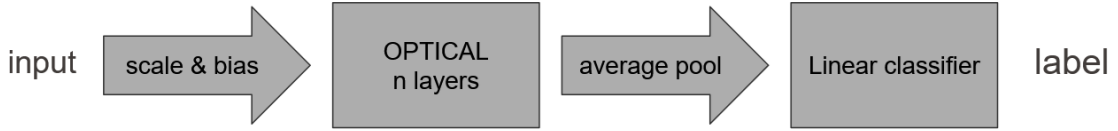


Figure 1.2: Optical neural network pipeline: input linear transformation via scale and bias on multiple optical layers, average pooling, and a linear classifier at the end.

The network parameters scale and bias are learned using backpropagation through the optical model. While this approach is functional, training deeper optical networks remains challenging. Increasing the number of optical layers does not necessarily lead to better performance as a result of the large number of parameters and the resulting tendency to overfit.

This performance plateau is visible on the graphs in Figure 1.3. This motivates the exploration of alternative representations that would enable deeper optical architectures while maintaining good learning behavior.

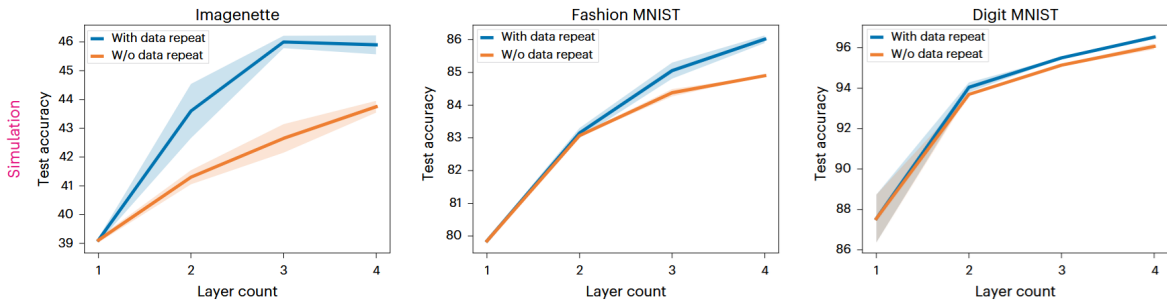


Figure 1.3: Performance plateau with growing layer number, results from [2].

Goals and approach

The goal of this project is to improve learning performance as the number of optical layers increases. To achieve this, we propose replacing direct optimization of the per-pixel scale and bias masks with a generative model that produces these parameters implicitly.

Instead of learning each mask value independently, a neural network generator will map spatial coordinates to parameter values. This implicit representation is expected to smooth out the optimization landscape and reduce the number of learnable parameters. As a result, deeper optical networks can be trained more reliably. The proposed architecture is visible in Figure 2.1 below.

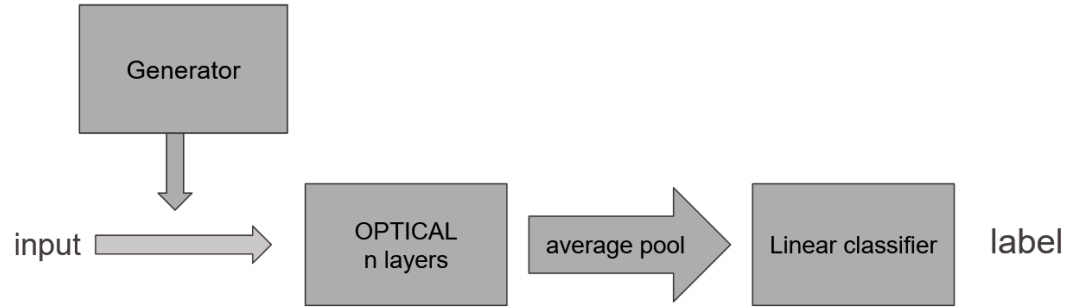


Figure 2.1: Proposed architecture: a generator produces scale and bias parameters that mixed with the input and then applied to the optical layers.

2.1 SIREN: Implicit Neural Representations

To implement the generator, we adopted the SIREN architecture (Implicit Neural Representations with Periodic Activation Functions) [1]. SIREN is a neural network that uses sinusoidal activation functions of the form

$$\sin[\omega_0(ax + a_0)]$$

where ω_0 is the base frequency and $s \cdot x + b$ is a linear transformation of the data from the previous layers or the input for the first layer.

SIRENs are particularly well suited for representing natural signals such as images and 3D shapes, and they have been shown to accurately model not only function values but also preserve their first and second-order derivatives. This makes them attractive for physics-related problems, where smoothness and differentiability are important, and makes them also promising in our application. [1]

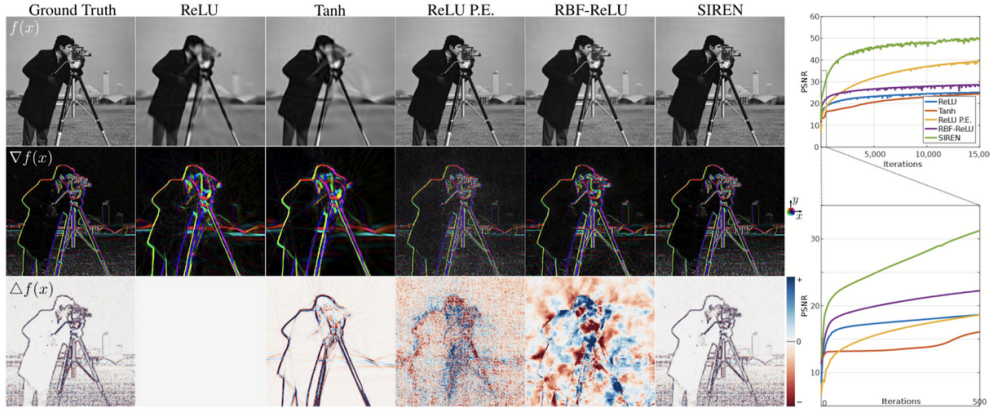


Figure 2.2: Comparison of representations in the first row, produced by different activation functions, with the first and second derivative of the image in the second and third row. Image from [1]

2.2 Architecture

2.2.1 Generator Network Design

In our implementation, the SIREN generator takes 2D spatial coordinates, corresponding to pixels in the scale and bias masks, as input and outputs one scale/bias value. In order to generate the whole scale/bias masks, the generator is then executed for every pixel in the corresponding mask. With this approach, we can get a smooth representation of the masks inside the generator and discretize it to individual pixels only at the end. This should lead to a smoother optimization landscape, a reduced number of parameters, and the scale/biases can be generated in the same fashion independent of the physical size of the optical SLM.

We tested several different SIREN network architectures, in the end settling on two hidden layers, each with 128 neurons. Input layers will consist of two features (XY coordinates) and the output is one scalar value for the given mask. The diagram of the whole architecture

is shown in Figure 2.3

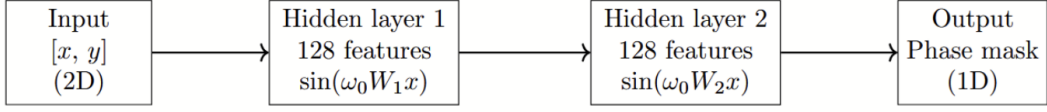


Figure 2.3: Detailed architecture of the SIREN generator and its integration with the optical network.

2.3 Optimization and Training

2.3.1 Hyperparameter Sensitivity

Training SIREN-based models was found to be very sensitive to setting hyperparameters, particularly base frequency and learning rate.

The base frequency ω_0 should approximately correspond to the frequencies found in the data set, to ease the training. The effect of varying the base frequency on the generated scale masks can be seen in Figure 2.4. The original paper [1] reported that $\omega_0 = 30$ worked for all the applications they tested. However, in our experiments, values in the range $\omega_0 \in [200, 600]$ yielded the best results, and we chose $\omega_0 = 400$ for all the experiments.

The learning rate was also found to be tightly coupled to the base frequency, with an approximate relationship

$$\text{learning rate} \approx \frac{C}{\omega_0}$$

where C is an empirical constant, that had to be found every dataset independently. Setting the correct learning rate was crucial for making the network learn in a reasonable amount of time.

A learning-rate scheduler was introduced, which would reduce the learning rate by a factor of ten over ten epochs. The effect of this is improved learning in later epochs, cutting the overall training time roughly in half.

While testing various models and settings of the model, we would firstly conduct experiments were with smaller optical layer sizes before scaling up to full resolution. As the network would have to simulate the entire optical setup in each optimization step, reducing the resolution of the physical system helped us to find the correct parameters. Here we used the fact, that the SIREN will always be same, no matter the required resolution, making it easily transferable among different physical setups.

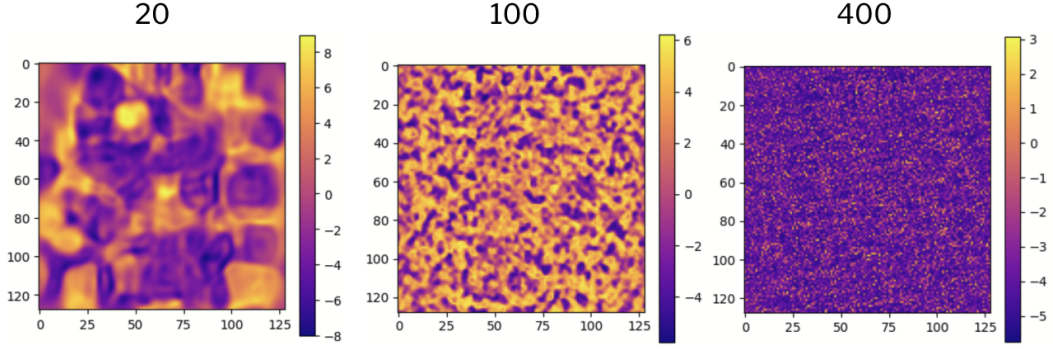


Figure 2.4: Effect of the base frequency ω_0 on the generated scale masks.

2.4 Data Processing

2.4.1 Dataset Description

We used the CIFAR-10 dataset, which is a popular benchmark for image classification, consisting of 60'000 small images of size 32×32 pixels divided into 10 object classes (such as airplanes, cars, animals, and ships). It contains 50'000 training images and 10'000 test images, with a balanced number of samples per class.

Although CIFAR-10 images are originally provided in RGB format, in this work we converted them to grayscale for simplicity. Preliminary test show that there is no limitation of how many channels can be processed, provided we generate a scale mask for each of the channels, but we wanted to remain comparable to the [2] paper.

2.4.2 Data Augmentation and Normalization

In order to further reduce the overfitting, that appeared while using multiple SIREN generators, data augmentation techniques such as random flips and crops were applied during training. Additionally, careful normalization of the input data was needed for achieving good performance.

Results

3.1 Baseline Performance

Baseline optical networks, described in the introduction part of this report, trained using pure backpropagation achieved accuracies of approximately 47%–50%, depending on optical resolution.

In the baseline model, each pixel size is optimizable on its own, so for an optical layer size of 512×512 , we get $512 \times 512 = 262'144$ optimizable parameters.

3.2 Our results

By contrast, the proposed SIREN-based generator achieved higher accuracy while using significantly fewer learnable parameters. For higher layer counts, the performance gain increased by approximately 4%, demonstrating that the generative approach scales more favorably with depth. The size of the SIREN is fixed, with the two hidden layers of 128 features each, this gives us about 128×128 parameters to optimize. This is a reduction of $16\times$ in comparison to the baseline model on the larger optical size.

The following Tables 3.1 and 3.2 compare the baseline model with the SIREN-based generator for different numbers of layers. All results are reported after training for 10 epochs with the learning rate scheduler for the SIREN model, and 20 epochs for the baseline model. We display results both for reduced-size optical size of 128×128 pixels and full 512×512 pixels, the size of the SIREN however stays constant and is independent of the optical layer size. The results are also visualized on a graph in Figure 3.1.

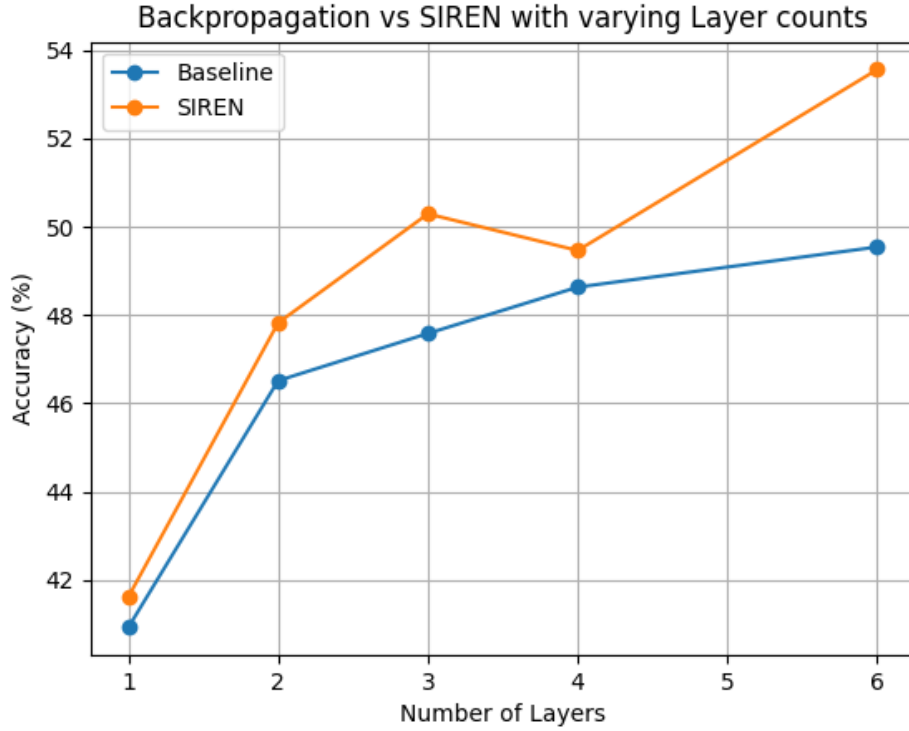


Figure 3.1: Comparison of classification accuracy between baseline backpropagation and the SIREN-based generator for varying numbers of optical layers.

3.3 Conclusion

This project demonstrates that generative programming using a SIREN generator can effectively improve the scalability of multilayer optical neural networks. The proposed approach reduces the parameter count while maintaining high optical resolution and improving classification performance. The results indicate that by enforcing structure through an implicit neural representation, the training performance could be improved. However, SIREN models are sensitive to hyperparameters and require a parameter search. Future work could explore if one generator can be pretrained and then subsequently used on different dataset, reducing the time required to train it from scratch.

Table 3.1: Optical size 128×128

Layers	Baseline (%)	SIREN (%)
1	39.66	40.94
2	44.19	46.61
3	45.55	45.74
4	46.79	48.90
6	47.28	50.91

Table 3.2: Optical size 512×512

Layers	Baseline (%)	SIREN (%)
1	40.92	41.61
2	46.51	47.82
3	47.58	50.29
4	48.63	49.46
6	49.54	53.56

Bibliography

- [1] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. “Implicit Neural Representations with Periodic Activation Functions”. In: *CoRR* abs/2006.09661 (2020). arXiv: 2006.09661. URL: <https://arxiv.org/abs/2006.09661>.
- [2] Mustafa Yildirim, Niyazi Ulas Dinc, Ilker Oguz, Demetri Psaltis, and Christophe Moser. “Non-linear processing with linear optics”. en. In: *Nat. Photonics* 18.10 (July 2024), pp. 1076–1082.