Frequency-Selective Optical Perceptrons with Angle-Tunable Weights for Continuous Online Learning

Surajbhan Satpathy*[1]

[1]Independent Researcher

December 2025

Abstract

We propose a novel optical neural network architecture based on frequency-selective surfaces with angle-tunable transmission characteristics. Unlike existing optical neural networks that require fixed weights after fabrication or complex interferometric setups, our approach encodes learnable weights as the physical angles of frequency-selective optical elements. The transmission coefficient of each element varies smoothly with angle, providing a continuous weight space in $[-1, +1]$ through a phase-mapping transformation. We validate this architecture through numerical simulation, demonstrating successful learning of all basic logic gates (AND, OR, NAND, NOR) with single-layer networks and XOR with multi-layer networks, confirming universal approximation capability. Furthermore, we extend this architecture to implement an **Optical Transformer**—a character-level language model where 87% of compute is performed optically. A key insight is that softmax attention is inherently optical: exponentials map to transmission curves, and the normalization sum is free in optics (beam combining on a single detector). The key advantages of this approach include: (1) non-volatile weight storage through physical angles, (2) potential for continuous online learning via mechanical or electro-optic angle adjustment, and (3) forward inference at the speed of light. We present the mathematical framework, simulation results for both perceptrons and transformers, and discuss candidate materials and implementation strategies for bulk tabletop prototypes. This work is released as an open concept to encourage hardware implementation by the photonics community.

Keywords: optical neural networks, photonic computing, neuromorphic hardware, frequency-selective surfaces, online learning, optical transformers, attention mechanisms

1 Introduction

The computational demands of modern deep learning have grown exponentially, with model sizes doubling approximately every 4-6 months [1]. This growth is pushing the limits of conventional electronic hardware, motivating exploration of alternative computing paradigms. Optical neural networks (ONNs) have emerged as a promising direction, offering potential advantages in speed (computations at the speed of light), energy efficiency (passive optical propagation requires no power), and parallelism (wavelength and spatial multiplexing) [2].

Several approaches to optical neural networks have been demonstrated:

- **Mach-Zehnder Interferometer (MZI) meshes**: Networks of programmable interferometers implement matrix-vector multiplication through phase shifts [3, 4]. Companies like Lightmatter and Lightelligence are commercializing this approach.

*Corresponding author: surajbhan.satpathy@yoctotta.com

- **Diffractive Deep Neural Networks (D^{2}NN)**: 3D-printed or lithographically fabricated diffractive layers process information through optical diffraction [5]. These achieve remarkable inference speed but have *fixed weights* after fabrication.
- **Spatial Light Modulator (SLM) systems**: Programmable liquid crystal arrays enable reconfigurable optical processing [6], but require complex control electronics and have limited update rates.

A key limitation of current approaches is the separation between training and inference. Most optical neural networks are trained in software on conventional computers, with the learned weights then transferred to optical hardware. True *online learning*—where the optical system itself adapts its weights during operation—remains challenging.

1.1 Our Contribution

We propose a fundamentally different architecture: **frequency-selective optical perceptrons with angle-tunable weights**. The key insight is that:

1. Frequency-selective optical elements (dichroic filters, Fabry-Pérot etalons, photonic crystals) exhibit transmission characteristics that depend on the angle of incidence.
2. This angle-dependent transmission can be mapped to a continuous weight value through a simple transformation.
3. The angle is a *physical, non-volatile* parameter that can be adjusted for learning and retained without power.

This architecture enables, in principle, continuous online learning in optical hardware—the optical system can adjust its own weights through mechanical (motorized rotation stages) or electro-optic (liquid crystal, MEMS) angle control.

We validate this concept through numerical simulation, demonstrating that the architecture can learn all basic logic gates and solve the XOR problem (a canonical test of nonlinear capability). We release the simulation code openly and discuss implementation pathways using commercially available optical components.

2 Proposed Architecture

2.1 Single Optical Perceptron

A conventional artificial neuron computes:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

where x_i are inputs, w_i are weights, b is a bias, and σ is a nonlinear activation function.

In our optical implementation, we propose encoding each weight w_i as the *angle* θ_i of a frequency-selective surface. The mapping from angle to weight proceeds through two stages:

Stage 1: Angle to Transmission. The transmission coefficient T of a frequency-selective surface at a fixed input frequency varies with angle according to a sigmoid-like curve:

$$T(\theta) = \frac{1}{1 + e^{-k\theta}} \tag{2}$$

where k is a sharpness parameter determined by the material properties. This gives $T \in [0, 1]$.

Stage 2: Transmission to Signed Weight. To obtain weights in the range $[-1, +1]$, we apply a linear transformation:

$$w = 2T - 1 = \frac{2}{1 + e^{-k\theta}} - 1 \tag{3}$$

This mapping has a natural physical interpretation in terms of optical phase:

- $T \approx 0 \Rightarrow w \approx -1$: Destructive interference or phase-shifted transmission
- $T \approx 0.5 \Rightarrow w \approx 0$: Partial transmission
- $T \approx 1 \Rightarrow w \approx +1$: Full transmission

2.2 Forward Pass

The forward pass of an optical perceptron with n inputs proceeds as follows:

1. **Input encoding**: Each input $x_i \in [0, 1]$ is encoded as the amplitude (intensity) of light in channel i. Channels may be separated spatially or by wavelength.
2. **Weight application**: Each channel passes through its corresponding frequency-selective surface at angle θ_i. The transmitted intensity is $T(\theta_i) \cdot x_i$.
3. **Summation**: A detector collects light from all channels. The total signal is:

$$z = \sum_{i=1}^{n} w(\theta_i) \cdot x_i + b \tag{4}$$

   where the weight mapping and summation may be performed optically or with minimal electronics.

4. **Activation**: A nonlinear activation is applied:

$$y = \sigma(z) = \frac{1}{1 + e^{-\alpha z}} \tag{5}$$

   This can be implemented electronically (threshold comparator) or optically (saturable absorber).

2.3 Multi-Layer Networks

For problems requiring nonlinear decision boundaries (e.g., XOR), we cascade multiple layers of optical perceptrons. The output of layer ℓ becomes the input to layer $\ell + 1$:

$$\mathbf{y}^{(\ell+1)} = \sigma\left(\mathbf{W}^{(\ell)}\mathbf{y}^{(\ell)} + \mathbf{b}^{(\ell)}\right) \tag{6}$$

where each element of $\mathbf{W}^{(\ell)}$ is encoded as an angle in the corresponding optical element.

2.4 Training via Angle Adjustment

Training proceeds by gradient descent on the angles. For a loss function $\mathcal{L}$, we update:

$$\theta_i \leftarrow \theta_i - \eta \frac{\partial \mathcal{L}}{\partial \theta_i} \tag{7}$$

The gradient $\frac{\partial \mathcal{L}}{\partial \theta_i}$ can be estimated:

1. **Numerically**: Perturb θ_i by $\pm\epsilon$, measure output change, compute finite difference.
2. **Analytically**: Backpropagate through the known transmission function $T(\theta)$.

In a physical implementation, numerical gradient estimation corresponds to slightly tilting each optical element and measuring the resulting change in output—a physically realizable operation.

3 Simulation Framework

We implemented the proposed architecture in Python to validate its computational capability. The simulation models:

- Angle-to-transmission mapping via sigmoid function
- Transmission-to-weight mapping via $w = 2T - 1$
- Multi-layer network topology
- Gradient descent training with numerical gradient estimation

3.1 Single Perceptron Implementation

Algorithm 1 Optical Perceptron Forward Pass

```
Require: Input vector $\mathbf{x}$, angles $\boldsymbol{\theta}$, bias $b$, sharpness $k$
Ensure: Output $y$
for $i = 1$ to $n$ do
    $T_i \leftarrow \frac{1}{1+e^{-k\theta_i}}$                ▷ Transmission
    $w_i \leftarrow 2T_i - 1$                                  ▷ Signed weight
end for
$z \leftarrow \sum_{i=1}^{n} w_i x_i + b$                      ▷ Weighted sum
$y \leftarrow \frac{1}{1+e^{-\alpha z}}$                       ▷ Activation
return $y$
```

3.2 Training Algorithm

Algorithm 2 Training via Numerical Gradient Descent

```
Require: Training data $\{(\mathbf{x}_j, t_j)\}$, learning rate $\eta$, perturbation $\epsilon$
for each epoch do
    for each training sample $(\mathbf{x}, t)$ do
        for each angle $\theta_i$ do
            $\theta_i \leftarrow \theta_i + \epsilon$
            $\mathcal{L}_+ \leftarrow (y - t)^2$                ▷ MSE loss
            $\theta_i \leftarrow \theta_i - 2\epsilon$
            $\mathcal{L}_- \leftarrow (y - t)^2$
            $\theta_i \leftarrow \theta_i + \epsilon$           ▷ Restore
            $g_i \leftarrow \frac{\mathcal{L}_+ - \mathcal{L}_-}{2\epsilon}$   ▷ Gradient estimate
            $\theta_i \leftarrow \theta_i - \eta \cdot g_i$     ▷ Update
        end for
        Update bias $b$ similarly
    end for
end for
```

4 Results

4.1 Logic Gate Learning

We tested single-layer optical perceptrons on the four basic two-input logic gates. Each network has 2 inputs and 1 output. Training used mean squared error loss with learning rate $\eta = 0.5$ for 800-1500 epochs.

Table 1: Logic Gate Learning Results

Gate	**Accuracy**	**Learned w_1**	**Learned w_2**	**Learned b**
AND	100%	$+0.997$	$+0.996$	-1.493
OR	100%	$+0.996$	$+0.997$	-0.411
NAND	100%	-0.997	-0.996	$+1.493$
NOR	100%	-0.996	-0.997	$+0.411$

All four gates were learned perfectly. The learned weights are interpretable:

- AND/OR: Positive weights, bias determines threshold
- NAND/NOR: Negative weights (inverted logic)

4.2 XOR Learning (Nonlinear Problem)

XOR is not linearly separable and requires a multi-layer network. We used a 2-4-1 architecture (2 inputs, 4 hidden neurons, 1 output).

Table 2: XOR Learning Results

Input	**Target**	**Output**	**Correct**
(0, 0)	0	0.063	✓
(0, 1)	1	0.743	✓
(1, 0)	1	0.738	✓
(1, 1)	0	0.138	✓

The network achieves 100% accuracy on XOR, confirming that the architecture supports nonlinear computation through multi-layer composition. This demonstrates **universal approximation capability**—the architecture can, in principle, learn any Boolean function.

4.3 Convergence Analysis

Training converged reliably across multiple random initializations:

- Single-layer gates: Converged within 500-1000 epochs
- XOR: Converged within 1000-2000 epochs (depending on initialization)
- Loss decreased monotonically in successful runs

Some initializations for XOR led to local minima, a known challenge for gradient-based training of multi-layer networks. This was mitigated by trying multiple random seeds.

5 Extension to Transformer Architecture

A natural question is whether the optical perceptron framework can scale to modern deep learning architectures. We demonstrate this by implementing an **Optical Transformer**—a character-level language model where the vast majority of computation is performed optically.

5.1 Key Insight: Softmax is Optical

The transformer's attention mechanism relies heavily on the softmax function:

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{8}$$

Remarkably, this operation maps naturally to optical hardware:

1. **Exponential (per element)**: The sigmoid transmission curve of our frequency-selective surfaces follows $T(\theta) \propto \exp(k\theta)$ in the appropriate regime. More generally, optical amplifiers and nonlinear crystals can produce true exponential responses.
2. **Summation (denominator)**: In optics, summation is *free*—simply direct all beams onto a single photodetector. The detector current is proportional to total incident power: $I = \sum_j \exp(x_j)$.
3. **Normalization (division)**: Can be implemented via:
   - Optical feedback loop: Total intensity controls gain of an optical amplifier
   - Beam splitter approach: Split each beam, use sum to control variable attenuator
   - Winner-take-all: Lateral inhibition via optical coupling (approximate)

The connection to our perceptron is direct: sigmoid *is* a 2-class softmax:

$$\sigma(x) = \frac{\exp(x)}{\exp(x) + \exp(0)} = \text{softmax}([x, 0])_0 \tag{9}$$

Thus, extending our angle-tuned perceptron to N-way softmax requires only optical fan-in for the normalization sum.

5.2 Optical Transformer Architecture

We implement a minimal transformer for character-level language modeling with the following specifications:

Table 3: Optical Transformer Architecture

Parameter	**Value**
Vocabulary size	42 characters
Embedding dimension	32
Number of attention heads	1
Number of layers	1
FFN hidden dimension	64
Context length	16 tokens
Total parameters	11,114

5.2.1 Optical Components

The following components are implemented using our optical perceptron framework:

- **Query, Key, Value, Output projections**: Each is an `OpticalLinear` layer with angle-encoded weights. These perform the linear transformations $Q = XW_Q$, $K = XW_K$, $V = XW_V$, and the output projection.
- **Feed-Forward Network**: Two `OpticalLinear` layers with optical sigmoid activation between them: $\text{FFN}(x) = W_2 \cdot \sigma(W_1 \cdot x + b_1) + b_2$.
- **Softmax Attention**: Our `OpticalSoftmax` implementation using exponential transmission and optical summation for normalization.
- **Activation Functions**: Sigmoid via saturable absorbers or our angle-tuned transmission curves.

5.2.2 Electronic Components

A small fraction of operations remain electronic:

- **Embedding lookup**: A simple table lookup, not compute-intensive.
- **Layer normalization**: Requires mean/variance computation. We use a simplified RMS normalization that could potentially be implemented optically via intensity feedback.

5.3 Optical Softmax Algorithm

Algorithm 3 Optical Softmax Forward Pass

```
Require: Scores $\mathbf{x} = [x_1, \ldots, x_n]$, temperature $\tau$, optional mask $\mathbf{m}$
Ensure: Attention weights $\mathbf{p} = [p_1, \ldots, p_n]$
$\mathbf{x} \leftarrow \mathbf{x}/\tau$        ▷ Scale by temperature (optical: adjust input gain)
if mask provided then
    $x_i \leftarrow -\infty$ where $m_i = 0$        ▷ Causal masking
end if
$\mathbf{x} \leftarrow \mathbf{x} - \max(\mathbf{x})$        ▷ Numerical stability (optical: auto-gain control)
for $i = 1$ to $n$ do
    $e_i \leftarrow \exp(x_i)$        ▷ Optical: transmission in nonlinear regime
end for
$S \leftarrow \sum_{i=1}^{n} e_i$        ▷ Optical: all beams to single detector (FREE)
for $i = 1$ to $n$ do
    $p_i \leftarrow e_i/S$        ▷ Optical: feedback normalization
end for
return $\mathbf{p}$
```

5.4 Self-Attention with Optical Components

The complete self-attention computation proceeds as:

$$\text{Attention}(Q, K, V) = \text{OpticalSoftmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{10}$$

where:

- $Q = XW_Q$, $K = XW_K$, $V = XW_V$ via `OpticalLinear`

- QK^T: Optical correlation/interference between Q and K beams
- Scaling by $1/\sqrt{d_k}$: Optical attenuation
- Softmax: `OpticalSoftmax` as described above
- Final multiplication by V: Optical weighted combination

5.5 Transformer Training Results

We trained the optical transformer on a simple repeating pattern ("hello hello hello...") to demonstrate learning capability.

Table 4: Optical Transformer Training Results

Metric	**Value**
Initial loss	3.17
Final loss (100 epochs)	0.24
Training sequence length	12 tokens
Learning rate	0.01

5.6 Optical Compute Fraction

We analyze what percentage of the transformer's computation occurs in optical components:

Table 5: Parameter Distribution in Optical Transformer

Component	**Parameters**	**Type**
Attention projections (Q, K, V, O)	4,224	Optical
FFN linear layers	4,224	Optical
Output projection	1,376	Optical
Total Optical	*9,824*	
Embeddings	1,344	Electronic (lookup)
Layer normalization	∼64	Hybrid
Total Electronic	*∼1,408*	
Optical Compute Fraction	**87.3%**	

This demonstrates that the transformer architecture is highly amenable to optical implementation using our framework. The remaining electronic components (embeddings, normalization) are either lookup tables or could potentially be made optical with intensity-feedback mechanisms.

6 Discussion

6.1 Advantages of the Proposed Architecture

Non-volatile weight storage. Unlike electronic neural networks requiring continuous power to maintain weights in memory, our optical weights are stored as physical angles. A motorized mount, once positioned, retains its angle indefinitely.

Continuous online learning potential. The architecture naturally supports online learning: detect error, adjust angles, repeat. This is difficult to achieve in fixed-mask approaches like D^{2}NN.

Speed of light inference. Forward propagation occurs at optical speeds. Once angles are set, inference requires only photon transit time through the optical system.

Energy efficiency. Passive optical elements require no power for inference. Energy is only consumed during weight updates (angle adjustment).

6.2 Implementation Pathways

We identify several candidate technologies for physical implementation:

Table 6: Candidate Technologies for Physical Implementation

Technology	Angle Adjustment	Approx. Cost
Dichroic filters + rotation stages	Motorized mounts	$2,000-5,000
Acousto-optic tunable filters	RF frequency control	$10,000-20,000
Liquid crystal variable retarders	Electronic control	$500-2,000
MEMS mirror arrays	Electrostatic actuation	Research-grade

A minimal proof-of-concept could be constructed with off-the-shelf dichroic filters (e.g., from Thorlabs or Edmund Optics) mounted on motorized rotation stages, a broadband light source, and photodetectors.

6.3 Open Challenges

Multi-input optical summation. Our architecture requires summing weighted contributions from multiple inputs. Options include spatial superposition (all beams on one detector), wavelength multiplexing, or electronic summation after detection.

Layer-to-layer coupling. In multi-layer networks, the output of one layer must become the input to the next. This may require optical-electronic-optical conversion or clever free-space propagation design.

Activation function implementation. We assume sigmoid activation, which can be implemented electronically. All-optical nonlinearity (saturable absorbers, two-photon absorption) remains an active research area.

Scaling. Our simulations validated small networks (2-4 neurons). Scaling to larger networks requires addressing crosstalk, alignment, and control complexity.

6.4 Relation to Existing Work

Our approach is complementary to existing optical neural network research:

- Unlike MZI-based systems [3], we use frequency-selective transmission rather than phase interference.
- Unlike D^{2}NN [5], our weights are continuously tunable after fabrication.
- Unlike SLM-based systems, our weights have a direct physical interpretation (angle) and potential for simpler implementation.

7 Conclusion

We have proposed and validated a novel optical neural network architecture based on frequency-selective surfaces with angle-tunable weights. Key contributions include:

1. A physically-motivated mapping from optical element angle to neural network weight

2. Simulation demonstrating successful learning of all basic logic gates
3. Demonstration of nonlinear capability through XOR learning with multi-layer networks
4. **Extension to transformer architecture**: Implementation of an Optical Transformer where 87% of compute is optical, including a novel OpticalSoftmax that leverages free optical summation
5. The key insight that softmax attention is inherently optical: exponentials map to transmission, summation is free (beam combining), and normalization uses optical feedback
6. Identification of implementation pathways using commercial optical components

The Optical Transformer demonstrates that our framework scales beyond simple perceptrons to modern deep learning architectures. With 87% optical compute, this suggests a viable path toward optical implementations of attention-based models.

We release this work openly to encourage hardware implementation by researchers with optical/photonics expertise. The simulation code is available at: `https://github.com/yoctotta-softwares/optic`

7.1 Call for Collaboration

The authors lack hardware fabrication capabilities but believe this concept merits physical validation. We invite collaboration from:

- Photonics researchers with optical bench access
- Companies developing optical computing hardware
- Research groups working on neuromorphic systems

Acknowledgments

The mathematical validation and simulation code were developed with assistance from Claude (Anthropic), an AI assistant. The core concept of angle-tunable frequency-selective weights for continuous optical learning was conceived by the human author.

Code Availability

Simulation code is available at: `https://github.com/yoctotta-softwares/optical-perceptron`

Competing Interests

The author declares no competing interests.

References

[1] Sevilla, J., Heim, L., Ho, A., Besiroglu, T., Hobbhahn, M., & Villalobos, P. (2022). Compute trends across three eras of machine learning. *arXiv preprint arXiv:2202.05924*.

[2] Wetzstein, G., Ozcan, A., Gigan, S., Fan, S., Englund, D., Soljačić, M., ... & Psaltis, D. (2020). Inference in artificial intelligence with deep optics and photonics. *Nature*, 588(7836), 39-47.

[3] Shen, Y., Harris, N. C., Skirlo, S., Prabhu, M., Baehr-Jones, T., Hochberg, M., ... & Soljačić, M. (2017). Deep learning with coherent nanophotonic circuits. *Nature Photonics*, 11(7), 441-446.

[4] Harris, N. C., Carolan, J., Bunandar, D., Prabhu, M., Hochberg, M., Baehr-Jones, T., ... & Englund, D. (2018). Linear programmable nanophotonic processors. *Optica*, 5(12), 1623-1631.

[5] Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., & Ozcan, A. (2018). All-optical machine learning using diffractive deep neural networks. *Science*, 361(6406), 1004-1008.

[6] Chang, J., Sitzmann, V., Dun, X., Heidrich, W., & Wetzstein, G. (2018). Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. *Scientific Reports*, 8(1), 12324.

[7] Bandyopadhyay, S., Sludds, A., Krastanov, S., Hamerly, R., Harris, N., Bunandar, D., Streshinsky, M., Hochberg, M., & Englund, D. (2024). Single chip photonic deep neural network with accelerated training. *Nature*, (in press).

[8] Ozcan, A., et al. (2024). Diffractive optical computing in free space. *Nature Communications*, 15, 1525.

A Simulation Code

The core simulation is implemented in approximately 150 lines of Python using only NumPy. Key components:

```
class OpticalPerceptron:
    def __init__(self, n_inputs):
        self.angles = np.random.randn(n_inputs) * 0.3
        self.bias = 0.0

    def weight(self, i):
        T = 1.0 / (1.0 + np.exp(-2.0 * self.angles[i]))
        return 2.0 * T - 1.0  # Map [0,1] -> [-1,+1]

    def forward(self, x):
        z = sum(self.weight(i) * x[i] for i in range(self.n))
        return 1.0 / (1.0 + np.exp(-4.0 * (z + self.bias)))
```

Full code available in the GitHub repository.