TÉCNICO LISBOA

Scientific Project

Digital-Optical Implementation of Quantum-Inspired and Classical Classifiers with a Joint Transform Correlator

João Guilherme

Supervisors:

Emmanuel Cruzeiro

Hugo Terças

May 19, 2025

Abstract

We explore the implementation of quantum-inspired and classical distance-based classifiers using an optical architecture known as the Joint Transform Correlator (JTC). These classifiers, including the classical and quantum nearest-mean classifier (CNM-C, QNM-C) and radial basis function classifier (RBF-C), rely on distance computations between a query sample and class prototypes. While traditionally executed in software, such operations become computationally intensive in high dimensions or large datasets. We propose a new optical implementation using a reflective, phase-only, spatial light modulator (SLM) and a CMOS camera within a 1-f Fourier setup, leveraging the speed and parallelism of coherent light. We detail the theoretical framework linking optical correlation with similarity measures used in classification, and describe the data encoding strategies that map feature vectors to optical patterns.

I. Introduction

Quantum-inspired algorithms are classical routines that use mathematical structures and physical intuitions from quantum mechanics — such as superposition-style probability amplitudes, interference or quantum state discrimination — yet run entirely on conventional hardware. These algorithms have multiple applications, including machine learning, where they promise faster convergence or reduced dimensional dependence compared with traditional methods.

Some machine-learning algorithms are based on distance. Here, the decision rule depends on the similarity between a query sample and a set of prototypes, typically evaluated through dot products or euclidean distance calculation. Although straightforward in software, these operations become a computational bottleneck for high-dimensional data streams or large datasets. There have been optical implementations of classical classification algorithms ([5], [3]). These implementations offer a compelling alternative: coherent light can perform calculations at the speed of propagation, exploiting spatial parallelism while consuming only milliwatts of power.

Among the various optical architectures, the *Joint Transform Correlator* (JTC) ([6], [2]) stands out for its simplicity and adaptability. Unlike classical VanderLugt systems ([8]), the JTC does not require a pre-fabricated filter; instead, the reference and query patterns are placed side by side in the input plane. A single Fourier transform—implemented with a lens—produces an output intensity whose off-axis terms encode the cross-correlation between the two patterns. When feature vectors are encoded as two-dimensional phase distributions, the correlation peak height provides a direct measure of their similarity, which can be mapped to a distance-based decision rule.

In this work we propose optical implementations of the Quantum-Inspired and Classical Nearest Mean Classifier ([1], [7]), and Radial Basis Function Classifier ([5], [3]), that use a single SLM and a 1-f lens system ([4]). We begin by reviewing the theoretical connection between optical correlation and amplitude-based similarity measures. We then describe our experimental set-up, which integrates a single reflective, phase only, spatial light modulator and a CMOS camera to implement a proof-of-concept optical classifier (Section **??**). Finally, we benchmark the models with the MNIST dataset and compare their performances with purely electronic implementations, highlighting the trade-offs in speed, energy, and classification accuracy (Section **??**).

II. Theory

i. Machine learning and supervised learning

Machine learning (ML) provides algorithmic tools that *learn* directly from data rather than relying on hand-crafted rules. In the *supervised* setting, learning is guided by *examples* whose correct answers are known in advance. We review the key ingredients used throughout this work.

Input data. We start from N *raw samples* $\mathbf{u}_i \in \mathbb{R}^p$, $i = 1, \ldots, N$, each accompanied by a *label* $y_i \in \{1, \ldots, K\}$. The labels partition the data into

$$C_k = \{\mathbf{u}_i : y_i = k\}, \qquad N_k = |C_k|, \qquad p_k = \frac{N_k}{N},$$

where N_k and p_k are the size and empirical probability of class k, respectively. It is convenient to organise the entire data set as a matrix $\mathbf{M} = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_N] \in \mathbb{R}^{p\times N}$ and the labels as a vector $\mathbf{y} = [y_1, \ldots, y_N]^\top$.

Training and test sets. A *training set* $(\mathbf{M}_{\text{train}}, \mathbf{y}_{\text{train}})$ is used to tune the parameters of a classifier, while a disjoint *test set* $(\mathbf{M}_{\text{test}}, \mathbf{y}_{\text{test}})$ gauges its ability to generalise to unseen data. The algorithm is trained on the training set and evaluated on the test set. In this work we will be using the MNIST dataset (Fig.1), with 60000 training samples and 10000 testing samples.

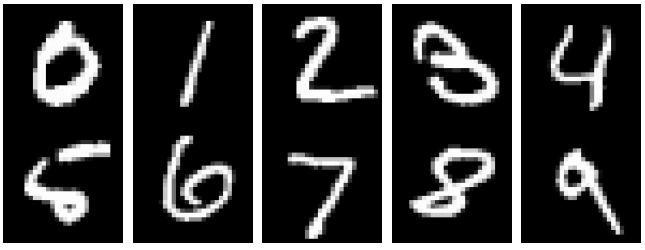


Figure 1: One representative example per class from the MNIST dataset. Each 28×28 grayscale image is flattened to a $p = 784$-dimensional feature vector before the pre-processing pipeline (centering, standardisation, PCA).

Pre-processing. Real-world data rarely arrive in a form that is optimal for learning. We apply three light transformations that are standard in pattern recognition:

(a) *Centring* removes global offsets, $\mathbf{u}_i \longmapsto \mathbf{u}_i - \bar{\mathbf{u}}$, where $\bar{\mathbf{u}} = N^{-1} \sum_{j=1}^{N} \mathbf{u}_j$.

(b) *Standardisation* rescales each feature so that every dimension has unit variance, preventing attributes with large numerical ranges from dominating the learning process.

(c) *Principal-component analysis* (PCA) projects the centred, standardised vectors onto the $p'(< p)$ directions of greatest variance, $\mathbf{x}_i = \Pi_{\text{PCA}} \mathbf{u}_i \in \mathbb{R}^{p'}$, thereby discarding redundant noise and shrinking the effective dimensionality.

Balanced accuracy. Performance is summarised through the *confusion matrix* $V_{k'k}$, whose entry counts how many test samples belonging to class k are classified as k'. The *balanced accuracy* (BA)

$$\text{BA} = \frac{1}{K} \sum_{k=1}^{K} \frac{V_{kk}}{N_k}$$

computes the mean of the per-class accuracies and is robust to class imbalance. When classes are homogeneous in size, BA reduces to the usual accuracy.

Classical Nearest-Mean Classifier (CNM-C). The nearest-mean classifier (NMC) is a simple, interpretable classifier that assigns a label to a test sample based on the *centroid* of the training samples in each class (Fig. 2). The *centroid* of class k is defined as the mean of the training samples in that class:

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{u}_i, \qquad \text{where } \mathbf{u}_i \in C_k.$$

The CNM classifier assigns a test sample $\mathbf{x}$ to the class j if:

$$d(\mathbf{x}, \boldsymbol{\mu}_j) = \min_{k=1,\ldots,K} d(\mathbf{x}, \boldsymbol{\mu}_k),$$

where the distance $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance. The CNM classifier is simple to implement, as it only requires the computation of the centroids and K distance calculations, one for each centroid.

Radial-Basis-Function Classifier (RBF-C). The RBF model represents a target function as a weighted sum of Gaussian basis functions, each centered at a specific location in the input space. These Gaussian functions produce localized responses that decrease with distance from their centers, allowing the model to approximate smooth and continuous mappings. By tuning the centers, widths, and weights of the basis functions, the RBF network can adapt to complex data distributions and interpolate between training samples with controlled generalization.

We can build a simple RBF network, shown in Fig.3, that contains an input layer of raw features, a hidden layer of M Gaussian units, and a linear output layer that produces one score per class. Every hidden unit returns a similarity value $\varphi_i(\mathbf{x}) = \exp\left[-\|\mathbf{x} - \mathbf{t}_i\|^2 / (2\sigma_i^2)\right]$ to its centre $\mathbf{t}_i$ with width σ_i, so the class score is a weighted sum $f_c(\mathbf{x}) = \sum_{i=1}^{M} w_{c,i}\, \varphi_i(\mathbf{x})$.

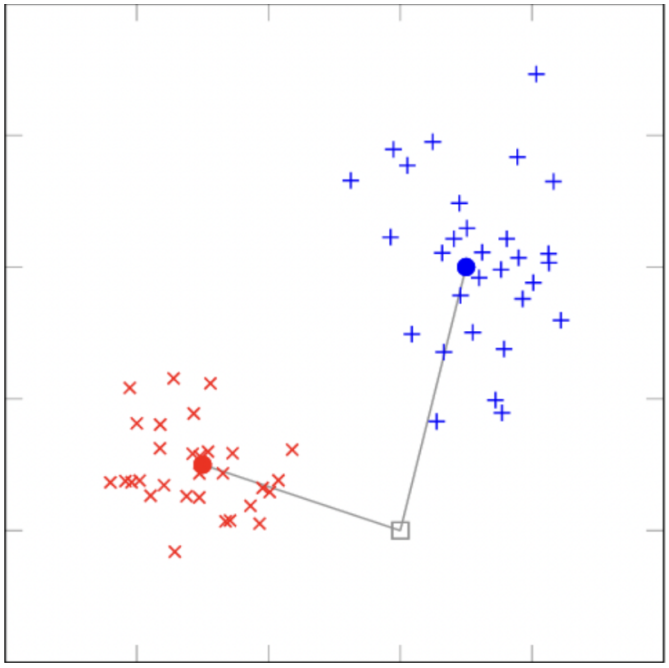


Figure 2: Ilustration of the CNM-C decision rule. The training set is split into two classes and the centroids are computed. The nearest centroid to the test sample is selected as the predicted class.

Training. Learning splits naturally into two parts:

(a) *Centre and width selection.* Mini-batch k-means clusters the training set; the cluster centroids become the RBF centres $\mathbf{t}_i$. Each width is set to $\sigma_i = k_\sigma \bar{d}_i$, where $\bar{d}_i$ is the average distance to the nearest neighbouring centre, with a floor $\sigma_{\min}$ to avoid underflow.

(b) *Output-weight estimation.* Defining the design matrix $\Phi_{n,i} = \varphi_i(\mathbf{x}_n)$, we solve the ridge-regularised normal equations $\mathbf{W} = (\Phi^\top \Phi + \lambda I)^{-1} \Phi^\top \mathbf{T}$, where $\mathbf{T}$ is the one-hot label matrix. This closed-form step replaces slow gradient-descent procedures and finishes in a single pass.

Inference. For a test vector $\mathbf{x}$, RBF-C computes the hidden activation vector $\boldsymbol{\varphi}(\mathbf{x}) = \left[\varphi_1(\mathbf{x}), \ldots, \varphi_M(\mathbf{x})\right]^\top$, then evaluates $\mathbf{f}(\mathbf{x}) = \mathbf{W}\boldsymbol{\varphi}(\mathbf{x})$ and assigns the class with a winner-takes-all rule: $c^\star = \arg\max_c f_c(\mathbf{x})$. Because the heavy computation is the M distance evaluations, we replace the Euclidean metric with the optical calculated distance, allowing these distances to be calculated faster and more energy-efficiently.

ii. Quantum-inspired classification

Quantum-inspired learning borrows mathematical objects native to quantum theory—*density operators*, *trace distance*, *superposition*—but applies them on ordinary CPUs/GPUs. By embedding classical feature vectors in a Hilbert space, one can exploit the geometry of quantum states to obtain alternative similarity measures that sometimes improve accuracy or robustness compared

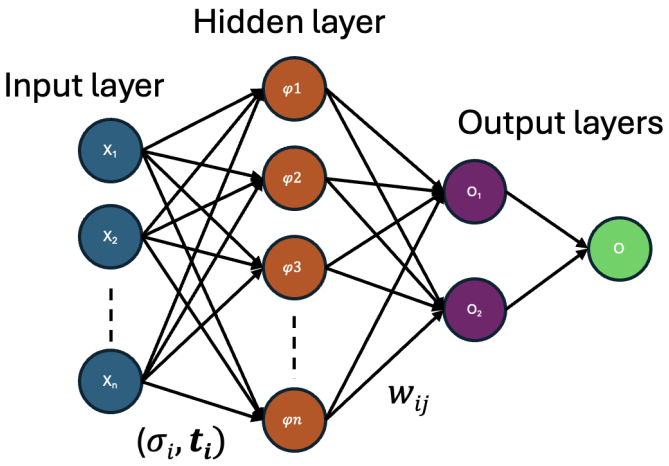


Figure 3: Architecture of the RBF network. The input layer feeds into a hidden layer of radial units, each centered at $\mathbf{t}_i$ with width σ_i. The output layer combines the gaussian responses through learned weights w_{ij}, followed by a winner-takes-all decision stage to determine the predicted class.

with purely classical methods [7]. Every quantum-inspired algorithm must start with a function that maps each classical data vector x to a density operator ρ_x. The choice of this encoding is very important, as it determines the properties of the resulting quantum-inspired algorithm.

Density–pattern encodings. All quantum-inspired classifiers require a map that sends a real feature vector $\mathbf{x} \in \mathbb{R}^d$ to a *density operator* $\rho_{\mathbf{x}} = \tilde{\mathbf{x}}\,\tilde{\mathbf{x}}^\top$. There are different ways of doing this mapping:

(a) Standard amplitude (SA) encoding.
Normalise the input vector to unit length:

$$\tilde{\mathbf{x}}^{(\mathrm{SA})} = \begin{cases} \dfrac{\mathbf{x}}{\|\mathbf{x}\|}, & \mathbf{x} \neq \mathbf{0}, \\ (1, 0, \ldots, 0)^\top, & \mathbf{x} = \mathbf{0}. \end{cases}$$

The encoded state remains in the original d-dimensional Hilbert space.

(b) Inverse stereographic (IS) encoding.
Embed $\mathbf{x}$ into $\mathbb{R}^{d+1}$ through the inverse stereographic projection onto the unit sphere:

$$\tilde{\mathbf{x}}^{(\mathrm{IS})} = \frac{1}{1 + \|\mathbf{x}\|^2} \Big(2x_1, \ldots, 2x_d, \|\mathbf{x}\|^2 - 1\Big)^\top.$$

This encoding adds an extra dimension to the input vector, which is then projected onto the unit sphere.

(c) Informative (INF) encoding [7].
Attach the norm as an extra amplitude so that both direction and length of $\mathbf{x}$ affect the representation:

$$\tilde{\mathbf{x}}^{(\mathrm{INF})} = \begin{cases} \dfrac{1}{\sqrt{2}} \Big(\dfrac{\mathbf{x}}{\|\mathbf{x}\|}, 1\Big)^\top, & \mathbf{x} \neq \mathbf{0}, \\ (0, \ldots, 0, 1)^\top, & \mathbf{x} = \mathbf{0}. \end{cases}$$

For any of the above encodings, the corresponding *density operator* is obtained by taking the outer product of the encoded vector with itself:

$$\rho_{\mathbf{x}} = \tilde{\mathbf{x}}\,\tilde{\mathbf{x}}^\top.$$

This guarantees $\rho_{\mathbf{x}}$ is positive semidefinite, unit-trace, and idempotent ($\rho_{\mathbf{x}}^2 = \rho_{\mathbf{x}}$), hence a valid pure quantum state. They differ in how much geometric information (direction, norm) is retained, a factor that can significantly affect classification performance.

Quantum centroid. For each class k with training subset $\mathcal{S}_k^{\mathrm{tr}}$, define its *quantum centroid* as the sum of the density operators from the class, divided by the number of elements in the class:

$$\varrho_k = \frac{1}{|\mathcal{S}_k^{\mathrm{tr}}|} \sum_{(\mathbf{x}_i, k) \in \mathcal{S}_k^{\mathrm{tr}}} \rho_{\mathbf{x}_i}.$$

Unlike the classical centroid, ϱ_k is usually a *mixed* state and has no direct counterpart in the original feature space.

Quantum distance measure. Similarity between two density patterns is quantified by the *trace distance*,

$$d_{\mathrm{tr}}(\rho, \sigma) = \tfrac{1}{2}\,\mathrm{Tr}\big|\rho - \sigma\big|,$$

where $|A| = \sqrt{A^\dagger A}$. Note that the trace distance defines a metric; it satisfies the triangle inequality, $d_{tr}(\rho, \sigma) \geq 0$ and $d_{tr}(\rho, \sigma) = d_{tr}(\sigma, \rho)$, and is equal to zero if and only if $\rho = \sigma$.

Having defined the quantum centroid and quantum distance, we can now define the quantum-inspired version of the nearest-mean classifier.

Quantum-inspired nearest-mean classifier (QNM-C). The quantum-inspired nearest-mean classifier (QNM-C) is the exact same as the CNM-C, but uses the quantum centroid and quantum distance instead of the classical centroid and Euclidean distance. The algorithm is as follows:

(a) Encode every training sample $\mathbf{x}_i$ as a density operator $\rho_{\mathbf{x}_i}$ using one of the encodings described in ii.

(b) Compute the quantum centroid for each class k:

$$\varrho_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \rho_{\mathbf{x}_i}.$$

(c) For each test density operator $\rho_{\mathbf{x}}$, assign the class k if

$$d_{\mathrm{tr}}(\rho_{\mathbf{x}}, \varrho_k) = \min_{j=1,\ldots,K} d_{\mathrm{tr}}(\rho_{\mathbf{x}}, \varrho_j).$$

Having defined both the classical and quantum-inspired nearest-mean classifiers, we can now describe the joint transform correlator (JTC) and how it can be used to implement these classifiers optically.

iii. Joint Transform Correlator

A *joint transform correlator* (JTC) is an optical architecture that performs the cross– and auto–correlation of two 2-D patterns by means of only *free-space propagation* and *Fourier transform* lenses, thereby exploiting the inherent parallelism and picosecond-scale speed of coherent light. Unlike classical Vander Lugt correlators, the JTC dispenses with a pre-fabricated matched filter: both the *reference* pattern $s_1(x, y)$ and the *query* pattern $s_2(x, y)$ are displayed side-by-side in the same input plane [4].

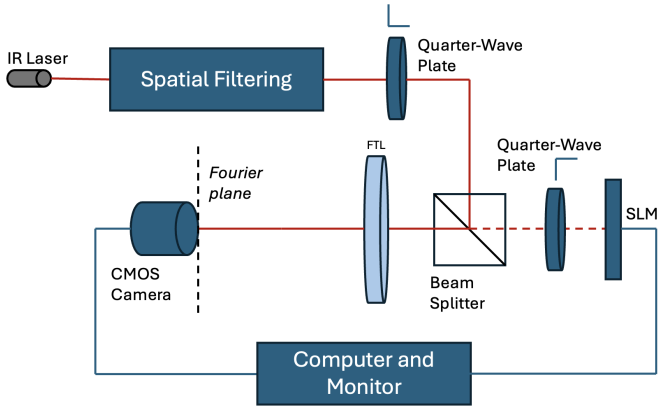


Figure 4: Optical setup for the joint transform correlator. The SLM displays the reference and query patterns side by side.

Operational principle. Let the two patterns be centred at $\pm x_0$ along the horizontal axis, i.e. $s_1(x + x_0, y)$ and $s_2(x - x_0, y)$. A $1f$-configuration lens (focal length f) produces at its back focal plane the joint Fourier transform

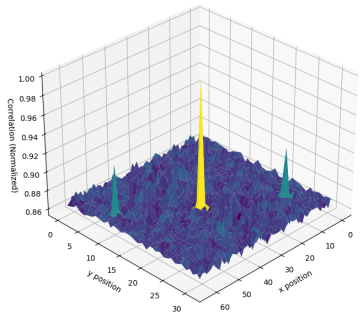
$$G(\alpha, \beta) = S_1\left(\tfrac{\alpha}{\lambda f}, \tfrac{\beta}{\lambda f}\right) e^{-i2\pi x_0 \alpha/\lambda f} + S_2\left(\tfrac{\alpha}{\lambda f}, \tfrac{\beta}{\lambda f}\right) e^{+i2\pi x_0 \alpha/\lambda f},$$

where (α, β) are spatial-frequency coordinates and S_j denotes the Fourier transform of s_j. A photosensor records the *joint power spectrum* $|G|^2$; its intensity contains four beating terms:

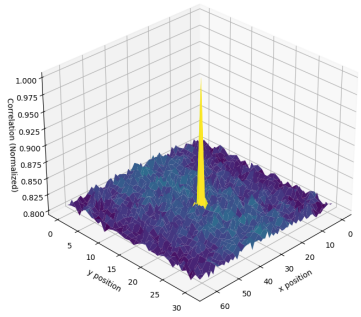
$$|G|^2 = |S_1|^2 + |S_2|^2 + S_1 S_2^* e^{-i4\pi x_0 \alpha/\lambda f} + S_2 S_1^* e^{+i4\pi x_0 \alpha/\lambda f}.$$

A second lens takes the inverse Fourier transform of this intensity pattern. The constant terms yield *autocorrelation* peaks near the optic axis, whereas the two cross-terms produce a pair of off-axis peaks located at $(\pm 2x_0, 0)$ whose heights are proportional to the cross-correlation $s_1 \star s_2$. Detecting the peak energy therefore reveals the similarity between the two input patterns.

TODO: INCLUDE A DIAGRAM OF THE JTC, AND THE CORRELATION PLANE, AS WELL AS SOME EXAMPLES OF CORRELATION (SHIFTED IMAGE VS ORIGINAL, ROTATED VS ORIGINAL, AND CORRELATION VALUES)



(a) Original image vs. shifted copy



(b) Original image vs. random image

Figure 5: Correlation-plane intensity produced by the joint transform correlator: (a) shows two strong off-axis peaks because the query is a shifted version of the reference, whereas (b) shows no significant peaks for an unrelated image.

Table 1: Classification accuracy (mean ± std.)

Classifier	Distance metric	Encoding	Accuracy (%)
RBF-C	Euclidean	-	-
RBF-C	JTC	-	-
CNM-C	Euclidean	—	80.38 ± 0.38
CNM-C	JTC	—	72.42 ± 0.55
QNM-C	Trace	Standard	85.84 ± 0.38
QNM-C	Fidelity	Standard	78.77 ± 0.34
QNM-C	Trace	Informative	81.26 ± 0.37
QNM-C	Fidelity	Informative	82.39 ± 0.40

iv. Optical implementations of the classifiers

III. Results

IV. Data Encoding

i. Gram Matrix Encoding

We consider a column vector u representing an image sample. To embed this into a higher-order representation, we construct the normalized outer product:

$$\rho = \frac{uu^T}{\mathrm{Tr}(uu^T)}$$

This yields a density-like matrix ρ with trace 1, encoding pairwise correlations between pixels. Such an encoding is inspired by quantum mechanical density operators and preserves structural information in the image.

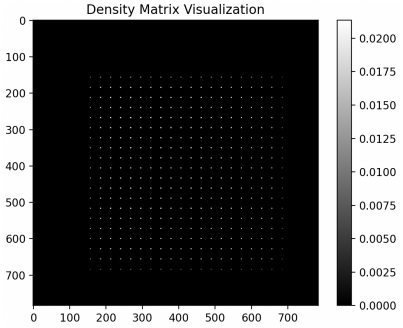


Figure 6: Gram matrix encoding ρ of the digit 1 image

V. Radial Basis Function Network

RBF networks model the hypothesis function $h(x)$ using a radial function centered around key points (called *centers*). Each center influences the prediction based on its proximity to the input x. The standard form is:

$$h(x) = \sum_{n=1}^{N} w_n \exp\left(-\gamma \|x - x_n\|^2\right)$$

where γ determines the width of the radial functions, and w_n are the learned weights.

i. Exact Interpolation

Given training data $D = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, we can formulate the interpolation condition:

$$\sum_{m=1}^{N} w_m \exp\left(-\gamma \|x_n - x_m\|^2\right) = y_n, \quad \forall n$$

In matrix notation:

$$\Phi \mathbf{w} = \mathbf{y}, \quad \text{where } \Phi_{n,m} = \exp\left(-\gamma \|x_n - x_m\|^2\right)$$

If Φ is invertible, we can directly solve:

$$\mathbf{w} = \Phi^{-1}\mathbf{y}$$

This yields an exact interpolating function.

ii. Classification

For classification, we apply a decision rule such as:

$$h(x) = \mathrm{sign}\left(\sum_{n=1}^{N} w_n \exp(-\gamma \|x - x_n\|^2)\right)$$

In multi-class settings, this can be extended to a softmax output over class scores.

iii. Training with Least Squares

Instead of enforcing exact interpolation, we may use a least squares approach to minimize:

$$E = \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

This leads to the ridge-regularized solution:

$$\mathbf{w} = (\Phi^T \Phi + \lambda I)^{-1} \Phi^T \mathbf{y}$$

where λ is a regularization parameter.

VI. Choosing RBF Centers

Using every data point as a center is computationally expensive. Instead, we reduce the number of centers to $K \ll N$ by clustering the dataset using **K-Means**:

We define:

$$J = \sum_{k=1}^{K} \sum_{x_n \in S_k} \|x_n - \mu_k\|^2$$

and use Lloyd's algorithm to iteratively minimize J:

$$\mu_k = \frac{1}{|S_k|} \sum_{x_n \in S_k} x_n$$
$$S_k = \{x_n : \|x_n - \mu_k\| \leq \|x_n - \mu_j\|, \ \forall j \neq k\}$$

Repeat these steps until convergence. Since the process is sensitive to initialization, we typically run K-Means multiple times and choose the best clustering.

Figure 7: Example of a Gaussian RBF centered at μ

VII. Implementation Notes

The RBF network was implemented in Python using NumPy and scikit-learn. To avoid memory issues, we:

- Use MiniBatchKMeans to find centers efficiently.
- Use perceptron-based or regularized least squares learning.
- Avoid computing large $\Phi^T \Phi$ matrices when possible.

VIII. Optical Im

IX. Conclusion

RBF networks provide an elegant and powerful framework for classification. Through Gaussian kernel construction and data-driven center selection, they can adapt to local structures in the data. When combined with efficient encodings such as Gram matrices and practical training algorithms, they scale to real-world tasks like digit recognition.

References

[1] E.Z. Cruzeiro, C. De Mol, S. Massar, et al. Quantum-inspired classification based on quantum state discrimination. *Quantum Machine Intelligence*, 6:79, 2024.

[2] Andrei Drăgulinescu. Optical correlators for cryptosystems and image recognition: A review. *Sensors*, 23(2):907, 2023.

[3] Wesley E. Foor and Mark A. Neifeld. Adaptive, optical, radial basis function neural network for handwritten digit recognition. *Applied Optics*, 34(32):7545–7555, 1995.

[4] Bahram Javidi and Joseph L. Horner. Single spatial light modulator joint transform correlator. *Applied Optics*, 28(5):1027–1032, 1989.

[5] M. A. Neifeld and D. Psaltis. Optical implementations of radial basis classifiers. *Applied Optics*, 32:1370–1379, 1993.

[6] D. Psaltis, E. G. Paek, and S. S. Venkatesh. Optical image correlation with a binary spatial light modulator. *Optical Engineering*, 23:698–704, 1984.

[7] G. Sergioli, G. Russo, E. Santucci, A. Stefano, S.E. Torrisi, S. Palmucci, C. Vancheri, and R. Giuntini. Quantum-inspired minimum distance classification in biomedical context. *quant-ph*, 2018.

[8] A. Vander Lugt. Signal detection by complex spatial filtering. *IEEE Transactions on Information Theory*, 10(2):139–145, Apr 1964.