UniTs - University of Trieste

Faculty of Data Science and Artificial Intelligence

Department of mathematics informatics and geosciences

Advanced Deep Learning and Kernel Methods

Lecturer:
Prof. Fabio Anselmi

Author:
Christian Faccio

September 30, 2025

github.com/christianfaccio christianfaccio@outlook.it

Preface

- **Kernel Methods**
- **Neural Tangent Kernel and Neural Networks**
- **Biologically inspired ANNs for Visual Cortex**
- **Geometric Deep Learning**
- **Adversarial Attacks and ANNs Implicit Bias**

Contents

1 Kernel Methods **1**
1.1 Math of Kernels 1
1.2 Blackboard 30/09/2025 1
1.2.1 Kernel PCA demonstration 1

1 Kernel Methods

1.1 Math of Kernels

1.2 Blackboard 30/09/2025

$$\alpha^* = \underset{\alpha \in \mathscr{R}^n}{\arg\min} ||y - K\alpha||_2^2 + \lambda \alpha^T K \alpha \quad [K]_{ij} = K(x_i, x_j)$$

then:

$$\frac{\partial}{\partial \alpha} ||y - K\alpha||_2^2 = \frac{\partial}{\partial \alpha_i} \alpha^T K \alpha \tag{1.1}$$

$$= \sum_q K_{iq} \alpha_q + \sum_{pq} \alpha_p K_{pi} \tag{1.2}$$

$$= [K\alpha]_i + \left[K^T\right]_{ip} \alpha_p \tag{1.3}$$

$$= 2\,[K_\alpha]_i \tag{1.4}$$

$$= \frac{\partial}{\partial \alpha} (y - K\alpha)^T (y - K\alpha) \tag{1.5}$$

$$= \frac{\partial}{\partial \alpha} \left(\underbrace{-\alpha^T K^T y - y^T K \alpha}_{-2\alpha^T K^T y} + \alpha^T K^T K \alpha \right) \tag{1.6}$$

$$= 2K^T (K\alpha - y) \tag{1.7}$$

so now

$$\frac{\partial}{\partial \alpha} = \frac{K}{n} (K\alpha - y) + \lambda K \alpha = 0 \tag{1.8}$$

$$= \frac{K}{n} (K\alpha - y + n\lambda \alpha) = 0 \tag{1.9}$$

and finally

$$\alpha = (k + n\lambda I)^{-1} y \tag{1.10}$$

1.2.1 Kernel PCA demonstration

suppose to have

$$\{x_i\}_{i=1}^n \quad x_i \in \mathscr{R}^d$$

and we want to find a projection on a lower dimensional space $\mathscr{R}^k$ with $k < d$. We calculate the covariance matrix:

$$C = \frac{1}{n} \sum_{i=1}^n x_i x_i^T = \frac{1}{n} X X^T \quad X = [x_1, \ldots, x_n]$$

and we want to find the eigenvalues and eigenvectors of C:

$$Cv_i = \lambda_i v_i$$

the usual trick is to create a map $\phi : \mathscr{R}^d \to \mathscr{H}$ and then compute the covariance matrix in $\mathscr{H}$:

$$C_\phi = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T$$

we want to solve the eigenvalue problem:

$$C_\phi v_i = \lambda_i v_i$$

so let's now prove that v can be expressed as a linear combination of the $\phi(x_i)$:

$$C_\phi v_i = \lambda_i v_i \tag{1.11}$$

$$\frac{1}{n}\sum_{i=1}^{n} \phi(x_i)\phi(x_i)^T v_i = \lambda_i v_i \tag{1.12}$$

$$\frac{1}{n}\sum_{i=1}^{n} \phi(x_i) \underbrace{\langle \phi(x_i), v_i \rangle}_{\alpha_i} = \lambda_i v_i \tag{1.13}$$

$$v_i = \frac{1}{n\lambda_i}\sum_{i=1}^{n} \alpha_i \phi(x_i) \tag{1.14}$$

we can now substitute this expression in the eigenvalue problem:

$$C_\phi v_i = \lambda_i v_i \tag{1.15}$$

$$\frac{1}{n}\sum_{j=1}^{n} \phi(x_j)\phi(x_j)^T \left(\frac{1}{n\lambda_i}\sum_{k=1}^{n} \alpha_k \phi(x_k)\right) = \lambda_i \left(\frac{1}{n\lambda_i}\sum_{k=1}^{n} \alpha_k \phi(x_k)\right) \tag{1.16}$$

$$\frac{1}{n^2\lambda_i}\sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_k \phi(x_j) \underbrace{\langle \phi(x_j), \phi(x_k) \rangle}_{K_{jk}} = \frac{1}{n}\sum_{k=1}^{n} \alpha_k \phi(x_k) \tag{1.17}$$

we can now multiply both sides by $\phi(x_i)$ and use the kernel trick:

$$\frac{1}{n^2\lambda_i}\sum_{j=1}^{n}\sum_{k=1}^{n} \alpha_k K_{ij} K_{jk} = \frac{1}{n}\sum_{k=1}^{n} \alpha_k K_{ik} \tag{1.18}$$

$$\frac{1}{n\lambda_i}\sum_{j=1}^{n} K_{ij} \underbrace{\left(\frac{1}{n}\sum_{k=1}^{n} K_{jk}\alpha_k\right)}_{(K\alpha)_j} = (K\alpha)_i \tag{1.19}$$

$$\frac{1}{n\lambda_i}(K^2\alpha)_i = (K\alpha)_i \tag{1.20}$$

so we have to solve the eigenvalue problem:

$$K\alpha_i = n\lambda_i \alpha_i$$

and then we can find the projection of a new point x as:

$$\begin{aligned}
\langle v_i, \phi(x) \rangle &= \left\langle \frac{1}{n\lambda_i} \sum_{j=1}^{n} \alpha_j \phi(x_j), \phi(x) \right\rangle && (1.21) \\
&= \frac{1}{n\lambda_i} \sum_{j=1}^{n} \alpha_j \underbrace{\langle \phi(x_j), \phi(x) \rangle}_{K(x_j,x)} && (1.22) \\
&= \frac{1}{n\lambda_i} \sum_{j=1}^{n} \alpha_j K(x_j,x) && (1.23)
\end{aligned}$$

Bibliography

[1] David Goldberg. "What every computer scientist should know about floating-point arithmetic". In: *ACM computing surveys (CSUR)* 23.1 (1991), pp. 5–48.

[2] *High Performance Computing — digital-strategy.ec.europa.eu*. https://digital-strategy.ec.europa.eu/en/policies/high-performance-computing.