Archaeology of Intelligent Machines: Convolutional Neural Networks vs. Convolutional Kernel Networks

Andrei Matei-Octavian
matei.andrei@s.unibuc.ro

Tapai Delia-Maria
delia-maria.tapai@s.unibuc.ro

Costache Carolina-Andreea
carolina-andreea.costache@s.unibuc.ro

Abstract

Our project investigates the performance and theoretical foundations of Convolutional Kernel Networks (CKN) compared to standard Convolutional Neural Networks (CNN). CKNs leverage the kernel trick within a Reproducing Kernel Hilbert Space (RKHS) to create robust representations. We evaluate both approaches across varying datasets, including MNIST, CIFAR-10, and multiple bioinformatics sequence datasets, assessing accuracy, stability, and computational resource efficiency.

1 Introduction

Automatic image and sequence recognition typically relies on Convolutional Neural Networks (CNN). We explore the hybrid approach of Convolutional Kernel Networks (CKN), where representation learning relies on the kernel trick rather than purely learned filters. This approach connects deep learning architectures with the non-linear feature mapping capabilities of kernel methods.

Contributions

Our work was divided as follows:

- Matei: He came up with the idea and done most of the research on the topic. Implementation of standard CNN baselines and four CKN variants: Linear, Polynomial, Spherical, and Gaussian on 2 distinct bio data sets.

- Delia: Implementation and evaluation of standard CNN baselines and four CKN variants: Linear, Polynomial, Spherical, and Gaussian on 2 distinct bio data sets.

- Carolina: Implementation and evaluation of CNN baselines and four CKN variants: Linear, Polynomial, Spherical, and Gaussian on 2 image data sets.

- Together: Stability assessment of models' performance over 100 independent runs (depending on the complexity of the data set). Comparative efficiency benchmarks for training and inference latency.

Motivation and Literature

We chose this topic to understand the mathematical underpinnings of why convolutions work. Deep Convolutional Representations extract increasingly complex features through layers. In contrast, RKHS focuses on a special mathematical space where data similarity is compared using kernels (Hui, 2023).

The seminal work by Mairal et al. (2014) introduces the CKN, arguing that the kernel trick simplifies operations by requiring only pairwise evaluation rather than explicit high-dimensional vector manipulation. This approach has been extended to other fields; for instance, Song et al. (2019) applied similar kernel methods in medical imaging where data volume is low, and Chen et al. (2019) utilized it in bioinformatics for sequence analysis.

2 Approach

Training was at fisrt performed using TensorFlow/Keras on Google Colab leveraging GPU acceleration. At some point we started to feel the limitations of the free version so we switched and trained the models on our own machines.

Code Repository: `https://github.com/matei21/CKN-Presentation/tree/main`

2.1 Data Preprocessing

Pixel intensity values were normalized to the range $[0, 1]$ to ensure numerical stability and gradient descent convergence. DNA and RNA sequences were processed for pattern recognition tasks.

2.2 Baseline CNN Architectures

MNIST: A sequential CNN with two layers (32 and 64 filters). **CIFAR-10:** A simple architecture composed of three small blocks (32, 64, 64 filters) followed by a dense block.

2.3 Convolutional Kernel Networks (CKN)

CKNs approximate kernel functions on local neighborhoods to achieve robustness to small transformations. We evaluated Linear, Polynomial, Spherical (normalized cosine similarity), and Gaussian (RBF) kernels.

3 MNIST Results

The study tested 100 runs per model to assess stability. Non-linear kernels (excluding Gaussian) achieved superior accuracy compared to the CNN.

Model	Mean Acc	Std Dev	Mean AUC
CNN	0.991	0.001	1.000
Linear	0.991	0.001	1.000
Polynomial	**0.992**	0.001	1.000
Spherical	**0.992**	0.001	1.000
Gaussian	0.504	0.432	0.725

Table 1: MNIST performance metrics across 100 runs.

The Linear CKN achieved a dramatic increase in inference speed, being twice as fast as the baseline CNN (262ms vs 526ms). However, the Gaussian CKN demonstrated catastrophic failure due to extreme instability (Std Dev 0.432).

4 CIFAR-10 Results

For CIFAR-10, the Spherical CKN achieved the highest mean accuracy at 63.45%.

The Linear CKN exhibited a remarkable inference time of only 47ms, making it more than twice as fast as the CNN baseline.

Model	Mean Acc	Train(s)	Infer(ms)
CNN	55.45%	1.83s	114ms
Linear	57.06%	2.11s	**47ms**
Polynomial	62.48%	2.37s	90ms
Spherical	**63.45%**	3.55s	145ms
Gaussian	62.29%	3.67s	148ms

Table 2: CIFAR-10 accuracy and efficiency.

5 Bioinformatics Studies

CKNs were applied to DNA and RNA classification tasks to identify functional sites.

5.1 Promoter Gene E-coli

The Polynomial CKN was the clear winner, achieving the highest Mean Accuracy (0.901) and Mean AUC (0.965).

Model	Mean Acc	Mean AUC	AUC Std
CNN	0.882	0.957	0.041
Linear	0.891	0.958	0.040
Polynomial	**0.901**	**0.965**	**0.038**
Spherical	0.858	0.932	0.055
Gaussian	0.891	0.956	0.044

Table 3: E-coli promoter dataset results.

5.2 ALKBH5 and PTBv1 Studies

For ALKBH5, Polynomial CKN achieved the highest Mean Accuracy (63.5%) and Mean AUC (0.7034). For PTBv1, all models reached near-perfect accuracy, but the Linear CKN was 35% faster than the CNN during inference.

5.3 m6A Binding Site Prediction

For m6A site prediction, the Polynomial CKN provided the best combination of maximum AUC (0.8250) and stability. The Linear CKN offered a 25% reduction in inference time compared to the CNN.

6 Conclusion

CKNs are a valid alternative to CNNs. Polynomial CKNs offer maximum performance and stability for biological discovery, while Linear CKNs provide game-changing inference speed advantages, performing calculations up to 35% faster than standard CNNs.

Model	Mean Acc	Mean AUC	Infer(ms)
CNN	0.747	0.8218	296.84
Linear	0.746	0.8222	**221.99**
Polynomial	**0.748**	**0.8250**	297.99
Spherical	0.746	0.8225	296.94
Gaussian	**0.748**	0.8233	323.09

Table 4: m6A binding site prediction results.

References

L. Chen et al. 2019. Kernel-based interactions for convolutional networks. *Bioinformatics*, 35(18):3294–3302.

Jonathan Hui. 2023. Reproducing kernel hilbert space for machine learning. `https://jonathan-hui.medium.com/reproducing-kernel-hilbert-space-for-machine-learning-8de67b6b5377`. Accessed: 2024-01-01.

Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. 2014. Convolutional kernel networks. In *Advances in neural information processing systems*, volume 27.

Y. Song et al. 2019. Approximate kernel learning for medical image analysis. *PubMed*.