Quantum Kernels

MA5770: Modelling Workshop

MD Karimulla Haque

Guided by,
Prof. A. K. B. Chand

February 18, 2025

Introduction to Quantum Machine Learning (QML)

- What is QML?
  - Uses quantum computing principles to enhance machine learning models.
  - Explores exponentially large Hilbert spaces for data representation.
- Why Quantum ML?
  - Classical ML struggles with high-dimensional problems.
  - Quantum computing provides a natural feature space expansion.
  - Potential to outperform classical models in certain tasks.

What is a Quantum Kernel?

- **Kernel trick:** Instead of explicitly mapping data into high dimensions, we use a kernel function to measure similarity. $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$
- Quantum kernel replaces classical kernel functions with a quantum feature map: $k_Q(x_i, x_j) = |\langle\psi(x_i)|\psi(x_j)\rangle|^2$

Feb 25:

(Learning Quantum Computing Basics + Kernel Methods)

- Learn the basics of quantum computing:
  - Qubits, quantum gates, superposition, and entanglement.
  - Parameterised quantum circuits (PQCs) and quantum measurement.
- Understand kernel methods in machine learning:
  - Classical Support Vector Machines (SVMs).
  - Kernel PCA
  - Kernel k-means
  - Raidal Basis Function
- The role of feature maps in kernel-based learning

(Understanding Neural Tangent Kernel (NTK) & Quantum Tangent Kernel (QTK))

- Study Neural Tangent Kernel (NTK) to understand why tangent kernels are useful in deep learning.
- Learn the definition and mathematical formulation of Quantum Tangent Kernel (QTK):
  - How QTK extends NTK to quantum circuits.
  - How overparameterization affects training in quantum models.
  - Why deep parameterized quantum circuits behave differently than conventional quantum kernels.
- Set up a Qiskit environment for experiments.

(Implementing Quantum Kernels and QTK Framework)

- Implement the conventional quantum kernel method as a baseline.
- Implement the Quantum Tangent Kernel (QTK) using parameterized quantum circuits.
- Test the QTK on synthetic datasets (e.g., small random datasets).

Apr 8:

(Extending QTK to Benchmark Datasets & Hyperparameter Tuning)

- Apply QTK to benchmark quantum datasets (e.g., MNIST with quantum embeddings).
- Compare QTK vs. conventional quantum kernel in terms of accuracy, efficiency, and scalability.
- Study the effect of hyperparameters (e.g., circuit depth, number of qubits) on QTK performance.

Apr 22:

(Generalization Analysis & Computational Complexity Study)

- Analyze the generalization properties of QTK compared to classical and quantum kernels.
- Investigate the computational complexity of computing QTK on quantum hardware.
- Identify practical bottlenecks and limitations.

May 6:

(Final Report & Submission)

- Summarise key findings and insights
- Prepare and submit a final detailed report

Thank You