Lesson 4.1: Parameterized States

Mathematical Intuition · Module 4 — Optimization Regularization

Node 4.1

Overview

Understanding parameterized quantum circuits (ansätze) as a way to explore the Hilbert space of quantum states.

Learning Objectives

- See ansätze as parameterized families of quantum states
- Understand how circuit depth and structure affect expressibility
- Recognize common ansatz types and their trade-offs
- Build intuition for the relationship between parameters and state space

Core Concepts

What Is an Ansatz?

Definition

- **Ansatz**: "educated guess" for the form of the solution
- **Parameterized circuit**: $\mathrm{U}(\theta)$ with tunable parameters $\theta = (\theta_1, \theta_2, \ldots, \theta_p)$
- **Trial state**: $|\psi(\theta)\rangle = \mathrm{U}(\theta)|0\rangle^n$ (starting from computational basis)
- **Goal**: find θ^* such that $|\psi(\theta^*)\rangle$ approximates ground state

Why Parameterize?

- **Exploration**: parameters let us search through state space
- **Optimization**: classical optimizer adjusts θ to minimize cost
- **Hardware efficiency**: shallow circuits reduce noise
- **Physical intuition**: structure can encode problem symmetries

Ansatz Structure

Single-Qubit Rotations

- $\mathbf{R}_x(\theta) = \mathbf{e\hat{}(-i\theta\sigma_x/2)}$: rotation around x-axis
- $\mathbf{R}_\gamma(\theta) = \mathbf{e\hat{}(-i\theta\sigma_\gamma/2)}$: rotation around y-axis
- $\mathbf{R}_u(\theta) = \mathbf{e\hat{}(-i\theta\sigma_u/2)}$: rotation around z-axis
- $\mathbf{U}_3(\theta,\varphi,\lambda)$: general single-qubit rotation

Entangling Gates

- **CNOT**: creates entanglement between qubits
- **CZ**: controlled-Z gate
- $\mathbf{RZZ}(\theta) = \mathbf{e\hat{}(-i\theta ZZ/2)}$: parameterized two-qubit gate

Layer Structure

- **Rotation layer**: apply $R_\gamma(\theta_i)$ to each qubit
- **Entangling layer**: apply CNOTs or other two-qubit gates
- **Repeat**: multiple layers increase expressibility

Common Ansatz Types

Hardware-Efficient Ansatz

- **Structure**: matches native gate set of quantum hardware
- **Pros**: shallow circuits, low noise
- **Cons**: may not capture problem structure
- **Example**: alternating single-qubit rotations and CNOTs

Unitary Coupled Cluster (UCC)

- **Structure**: based on quantum chemistry excitations
- **Pros**: physically motivated, systematic improvement
- **Cons**: deep circuits, many parameters
- **Example**: UCCSD (singles and doubles excitations)

QAOA-Inspired Ansatz

- **Structure**: alternating problem and mixer Hamiltonians
- **Pros**: good for combinatorial optimization
- **Cons**: requires problem-specific design
- **Example**: $e^{(-i\beta H\_mixer)}e$(-iγH_problem)

Symmetry-Preserving Ansatz

- **Structure**: respects problem symmetries (particle number, spin, etc.)
- **Pros**: reduces parameter space, improves optimization
- **Cons**: requires symmetry analysis
- **Example**: particle-conserving gates

Expressibility vs Trainability

Expressibility

- **How much of Hilbert space can the ansatz reach?**
- More parameters → more expressible
- Deeper circuits → more expressible
- Trade-off: expressibility vs circuit depth (noise)

Trainability

- **How easy is it to optimize the parameters?**
- Too many parameters → barren plateaus (vanishing gradients)
- Too few parameters → can't reach good solutions
- Sweet spot: enough expressibility, trainable gradients

Barren Plateaus

- **Problem**: gradients vanish exponentially with circuit depth
- **Cause**: random circuits have exponentially small gradients
- **Solutions**: problem-inspired ansätze, local cost functions, parameter initialization

Examples in Context

Quantum Chemistry

- **Ansatz**: UCC (unitary coupled cluster)
- **State**: $|\psi(\theta)\rangle = \mathrm{e}\hat{}(\mathrm{T}(\theta)\text{-}\mathrm{T}\dagger(\theta))|\mathrm{HF}\rangle$
- **Parameters**: excitation amplitudes
- **Goal**: find molecular ground state energy

Combinatorial Optimization

- **Ansatz**: QAOA (Quantum Approximate Optimization Algorithm)
- **State**: $|\psi(\beta,\gamma)\rangle = \prod_p \mathrm{e}^{(\text{-i}\beta_p \mathrm{H\_mixer})}\mathrm{e}(\text{-i}\gamma_p \mathrm{H\_cost})|+\rangle^n$
- **Parameters**: mixing angles β, cost angles γ
- **Goal**: find optimal solution to combinatorial problem

Quantum Machine Learning

- **Ansatz**: data-encoding + variational layers
- **State**: $|\psi(\mathrm{x},\theta)\rangle = \mathrm{U}(\theta)\mathrm{U\_encode}(\mathrm{x})|0\rangle^n$
- **Parameters**: trainable weights θ
- **Goal**: learn function $\mathrm{f}(\mathrm{x}) = \langle\psi(\mathrm{x},\theta)|\mathrm{M}|\psi(\mathrm{x},\theta)\rangle$

Designing an Ansatz

Step 1: Understand the Problem

- What symmetries does the problem have?
- What's the expected structure of the solution?
- What's the available quantum hardware?

Step 2: Choose Structure

- Hardware-efficient for NISQ devices
- Problem-inspired for better optimization
- Balance expressibility and trainability

Step 3: Initialize Parameters

- Random initialization can lead to barren plateaus
- Problem-inspired initialization (e.g., classical solution)
- Layerwise training

Step 4: Test and Iterate

- Check expressibility: can it reach good states?
- Check trainability: do gradients vanish?
- Adjust depth, structure, initialization

Practice Problems

1. How many parameters does a hardware-efficient ansatz with L layers and n qubits have?
2. Why might a UCC ansatz be better than a hardware-efficient ansatz for quantum chemistry?
3. What is a barren plateau, and how can you avoid it?

Key Takeaways

- Ansätze are parameterized quantum circuits that explore state space
- Structure matters: hardware-efficient vs problem-inspired
- Trade-off: expressibility (can reach good states) vs trainability (can optimize)
- Barren plateaus are a major challenge in deep variational circuits
- Good ansatz design requires understanding the problem and hardware

Next Steps

Next, we'll explore cost functions—how to define what we're trying to minimize in VQE.

Mathematical Intuition · Module 4 — Optimization Regularization · Node 4.1
Concepts: parameterized-circuits, hilbert-space, state-space, superposition

hbar.university — own.your.cognition