CSCI-739 Quantum Machine Learning

Homework 3

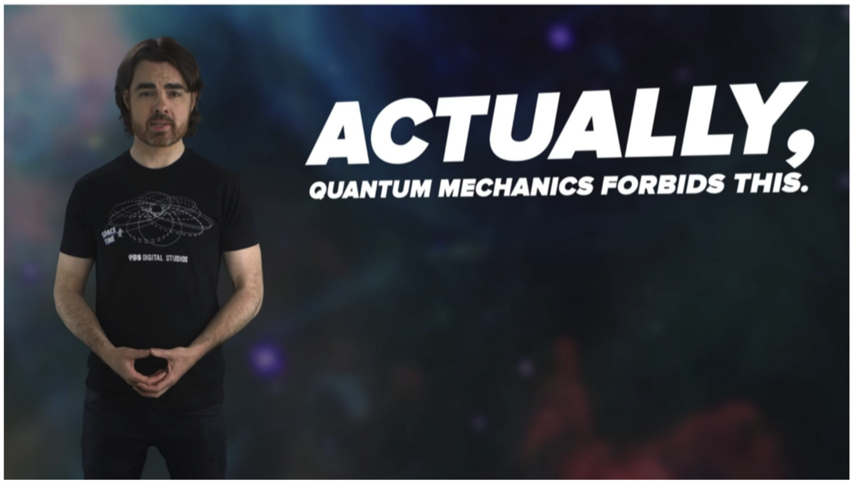
You got this, team. These problems are not arbitrary—they are designed as a journey through the essential pillars of QML. You will see how phase imperfections affect quantum operations (P1), how finding and approximating ground states links quantum physics to learning (P2), how the XOR problem illustrates quantum feature mappings and circuit expressivity (P3), and how quantum kernels power classification in QSVMs (P4). The bonus problems push you deeper: Bonus A reinforces scaling and measurement concepts, and Bonus B connects theory to optimization behavior and barren plateaus. Each step builds intuition that scales from single-qubit reasoning to multi-qubit learning architectures. If you can reason through these, you're already thinking like a quantum scientist. I believe in you all! ☺

uhum inspirational speech
Quantum machine learning is a blend of physics, computer science, and mathematics—a true frontier field. The first four problems of this homework each connect to a core competency we are developing in the course:

- *P1 (Phase-deformed X gates):* illustrates how coherent errors and gate deformations influence measurement outcomes, tying physical noise to computational results.
- *P2 (Ground states and VQCs):* demonstrates how energy minimization connects to learning and how variational quantum circuits approximate eigenstates.
- *P3 (XOR and encoding strategies):* shows how quantum encodings expand representational power beyond classical linear separability.
- *P4 (Quantum SVMs):* bridges classical ML concepts with quantum kernels, revealing how entanglement functions as a nonlinear feature map.
- *Bonus A/B:* expand your understanding of scaling, measurement variance, and gradient theory, preparing you for advanced QML topics like quantum optimization and barren plateau analysis.

Use lectures, notes, the internet and each other to research more on the problems and conquer this homework, team! Have fun ☺! Also:

The World: You cannot have fun learning quantum information sciences with Dr. Vogt



Problem 1 — A Phase-Deformed X Gate and Measurement Effects (25 pts)

Summary. We study a one-parameter family of unitaries $X(\varepsilon)$ that reduces to the Pauli-X gate at $\varepsilon = 0$, but deviates by a coherent (phase-induced) deformation when $\varepsilon \neq 0$. You will analyze measurement statistics for a single application ($n = 1$) and for repeated applications (n general), and reason about whether one can tune ε to achieve arbitrary measurement probabilities compared to the ideal case.

Define

$$X(\varepsilon) = \begin{pmatrix} \sin\varepsilon & \cos\varepsilon \\ \cos\varepsilon & -\sin\varepsilon \end{pmatrix} = \cos\varepsilon\, X + \sin\varepsilon\, Z.$$

(a) **Check the limits and unitarity.** Show that $X(0) = X$. Prove that $X(\varepsilon)$ is unitary (and Hermitian), and verify $X(\varepsilon)^2 = I$.

(b) **Special case $n = 1$ from $|0\rangle$.** Prepare $|0\rangle$ and apply $X(\varepsilon)$ once:

$$|\psi_1(\varepsilon)\rangle = X(\varepsilon)\,|0\rangle = \sin\varepsilon\,|0\rangle + \cos\varepsilon\,|1\rangle.$$

Compute the measurement probabilities in the computational basis:

$$P_\varepsilon^{(1)}(0), \qquad P_\varepsilon^{(1)}(1),$$

and verify their sum is 1. Compare the cases $\varepsilon = 0$ (ideal X) and $\varepsilon \neq 0$. *Question:* by choosing ε, can you achieve *any* target probability $P(1) \in [0, 1]$ in this single-application setting?

(c) **General n applications from $|0\rangle$.** Using $X(\varepsilon)^2 = I$, compute

$$|\psi_n(\varepsilon)\rangle = X(\varepsilon)^n\,|0\rangle,$$

hint: there is a result when n is even versus n is odd ☺. Compute $P_\varepsilon^{(n)}(0), P_\varepsilon^{(n)}(1)$ and compare to the ideal ($\varepsilon = 0$). *Question:* for general n, can you still set $P(1)$ arbitrarily by tuning ε (and therefore $P(0)$)? Discuss the parity effect.

(d) Comment on why the ideal versus the error-prone unitary operation is concerning, especially if we do not know exactly what ϵ could be in practice on noisy NISQ era hardware. Comment on why quantum error correction is therefore so critical.

Problem 2 — Ground State Analytically, Then via a VQC (25 pts)

Summary. This problem introduces the ground state (variational eigenvalue) problem. You will first compute the eigenvalues and eigenvectors of a simple 2×2 Hamiltonian analytically, then connect this to finding the minimal energy (ground state energy) and design a minimal variational quantum circuit (VQC) to reproduce that ground state through optimization.

Consider the one-qubit Hamiltonian:

$$H = \tfrac{1}{2}Z - X = \begin{pmatrix} \frac{1}{2} & -1 \\ -1 & -\frac{1}{2} \end{pmatrix}.$$

(a) **Eigenvalues and eigenvectors.** Find all eigenvalues and normalized eigenvectors of H. Write the eigenvectors explicitly in the computational basis $\{|0\rangle, |1\rangle\}$. Confirm that the matrix is Hermitian and that the eigenvectors form an orthonormal basis.

(b) **Ground state problem.** Define what is meant by the *ground state* and *ground state energy* of a Hamiltonian in quantum mechanics. Using your results from part (a), identify the ground state energy E_0 and the corresponding ground state $|\psi_0\rangle$.

(c) **Variational quantum circuit (VQC) approach.** Using the variational principle, propose a simple one-qubit ansatz such as

$$|\psi(\theta)\rangle = R_y(\theta)|0\rangle = \cos\tfrac{\theta}{2}|0\rangle + \sin\tfrac{\theta}{2}|1\rangle,$$

compute

$$C(\theta) = E(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle.$$

(i) *Analytic minimizer.* Derive $\theta_\star$ that minimizes $E(\theta)$ in closed form. Report $\theta_\star$ and $E(\theta_\star)$.

(ii) *Numerical VQE.* Implement a tiny VQE loop (e.g., grid search, Nelder–Mead, SPSA, or gradient descent) to minimize $E(\theta)$. Given the cost function $C(\theta)$ with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k) \in \mathbb{R}^k$ and gradient

$$\nabla_{\boldsymbol{\theta}} C = \left(\frac{\partial C}{\partial \theta_1}, \ldots, \frac{\partial C}{\partial \theta_k}\right)^{\top}.$$

Parameter-shift rule for one component:

$$\frac{\partial C(\theta)}{\partial \theta} = \frac{1}{2}\Big[C\big(\theta + \tfrac{\pi}{2}\big) - C\big(\theta - \tfrac{\pi}{2}\big)\Big].$$

Clarification: this is *one component* of $\nabla_{\boldsymbol{\theta}} C$. For a k-parameter circuit,

$$\frac{\partial C}{\partial \theta_i} = \frac{1}{2}\Big[C(\boldsymbol{\theta} + \tfrac{\pi}{2}\mathbf{e}_i) - C(\boldsymbol{\theta} - \tfrac{\pi}{2}\mathbf{e}_i)\Big].$$

Here, $\mathbf{e}_i$ denotes the standard basis vector in parameter space $\mathbb{R}^k$, i.e.,

$$\mathbf{e}_i = (0, 0, \ldots, 1, \ldots, 0)^\top,$$

with a single 1 in the i-th position and zeros elsewhere. It isolates the i-th parameter so that the shift applies only to θ_i while keeping all other parameters fixed.

Report: (1) the optimized parameter $\hat{\theta}$, (2) the achieved value $E(\hat{\theta})$, and (3) whether $\hat{\theta}$ (modulo 2π and global phase) matches your analytic $\theta_\star$.

(iii) *Measurement note.* Measure in the Z basis for Z, and insert a Hadamard before measurement for X—this rotation transforms X into Z so that the same measurement hardware can capture both components accurately.

Problem 3 — The XOR Problem and Encoding Strategies with Variational Circuits (25 pts)

Summary. This problem connects quantum data encoding strategies to a fundamental classical learning challenge: the XOR classification problem. The XOR dataset cannot be separated by a linear boundary, making it the smallest nonlinearly separable problem in machine learning. By exploring basis, angle, and amplitude encodings, you will see how different quantum encodings expand the representational capacity of variational quantum circuits (VQCs). For this problem use for example S=100 shots, or your favorite number ☺.

The XOR truth table is:

x_1	x_2	Output (XOR)
0	0	0
0	1	1
1	0	1
1	1	0

Classically, no single linear separator can distinguish the XOR outputs. A VQC, however, can achieve this through nonlinear feature maps or entanglement.

(a) **Basis encoding.** Encode (x_1, x_2) directly into the computational basis as $|x_1x_2\rangle$. Construct a two-qubit circuit of the form

$$|\psi(\boldsymbol{\theta})\rangle = \mathrm{CNOT}_{0\to1}\big(R_y(\theta_0) \otimes R_y(\theta_1)\big)\,|x_1x_2\rangle\,,$$

and train the parameters $\boldsymbol{\theta} = (\theta_0, \theta_1)$ to predict XOR outputs (you may define $P(1)$ as the model's output probability). Discuss whether this encoding allows linear separability after measurement. Note to produce state $|x_1x_2\rangle$, apply the appropiate X operations (or lack thereof) on initial state $|00\rangle$ to produce $|x_1x_2\rangle$ ☺.

(b) **Angle encoding.** Encode the features using rotational gates:

$$U(x_1, x_2) = \big(R_y(\pi x_1) \otimes R_y(\pi x_2)\big)\,\mathrm{CNOT}_{0\to1}.$$

Train a variational layer $V(\boldsymbol{\theta})$ to classify XOR. Compare expressivity to the basis encoding and discuss why the inclusion of entanglement changes the outcome.

(c) **Objective function.** In all encodings, the model output can be expressed as an expectation value of a measurement operator, e.g.

$$C(\theta) = L(\boldsymbol{\theta}) = \frac{1}{N}\sum_i \left(f_{\mathrm{VQC}}(x_i;\boldsymbol{\theta}) - y_i\right)^2,$$

where $f_{\mathrm{VQC}}(x_i;\boldsymbol{\theta}) = \langle\psi(x_i;\boldsymbol{\theta})|Z|\psi(x_i;\boldsymbol{\theta})\rangle$ represents the predicted value of the circuit for input (x_i, x_2) and y_i is the XOR label. Minimize $L(\boldsymbol{\theta})$ using classical optimization, reporting the final loss and learned parameters. Discuss convergence behavior and sensitivity to initialization.

(d) **From measurement to objective evaluation.** After each circuit execution, estimate the probability of measuring $|1\rangle$ on the readout qubit via shot counts n_1 out of S: $p = \frac{n_1}{S}$, compute $\langle Z\rangle = 1 - 2p$, and set $f_{\mathrm{VQC}} = \frac{1-\langle Z\rangle}{2} = p$. Substitute into $L(\boldsymbol{\theta})$ to evaluate the empirical loss.

(e) **(Optional) Run on real IBM NISQ hardware (shot sweep).** You may execute your VQC on an IBM Quantum backend. Try very small shot counts $S \in \{1, 10, 100\}$ per input to observe noise effects and compare to a statevector simulator (no shots). Explain why increasing S reduces variance in the empirical loss and improves stability.

(f) **Reflection.** Compare the three encoding methods. Which encoding best allows a shallow VQC to learn the XOR mapping? How does entanglement or nonlinear feature mapping provide a quantum analogue of the classical kernel trick? Relate your findings to the idea of quantum feature spaces used later in Quantum Support Vector Machines.

Problem 4 — Quantum Support Vector Machine (QSVM) for XOR (25 pts)

Summary. You will solve the XOR classification problem using a *quantum* kernel SVM. First confirm why a *linear* classical SVM fails on raw inputs. Then define a two-qubit feature map $U_\phi(x)$, estimate the quantum kernel matrix $K(x_i, x_j) = |\langle\phi(x_i)|\phi(x_j)\rangle|^2$, and train a classical SVM with this kernel (a QSVM). Finally, compare performance to your VQC from Problem 3. For this problem use for example 100 shots, or your favorite number ☺

Dataset (same ordering as Problem 3).

$$\mathcal{X} = \{(0,0), (0,1), (1,0), (1,1)\}, \quad y = \{-1, +1, +1, -1\}.$$

(a) **Why linear fails.** Show that no linear decision boundary in $\mathbb{R}^2$ separates XOR (a short geometric proof suffices). Optionally fit a linear SVM and report it cannot achieve 100% training accuracy.

(b) **Quantum feature map $U_\phi(x)$ (define the circuit).** Use two qubits initialized in $|00\rangle$. For input $x = (x_1, x_2) \in \{0,1\}^2$, define

$$U_\phi(x;\beta) = \left(R_z(\pi x_1) \otimes R_z(\pi x_2)\right) e^{-i\beta\, Z\otimes Z}, \qquad \beta = \tfrac{\pi}{4} \text{ (recommended)}.$$

Implement $e^{-i\beta Z\otimes Z}$ via

$$e^{-i\beta Z\otimes Z} = \mathrm{CNOT}_{0\to1}\left(I \otimes R_z(2\beta)\right)\mathrm{CNOT}_{0\to1},$$

and set $|\phi(x)\rangle = U_\phi(x;\beta)\,|00\rangle$.

(c) **Kernel estimation circuit (build K).** Define

$$K(x_i, x_j) = |\langle\phi(x_i)|\phi(x_j)\rangle|^2.$$

Estimate K with the inverse-composition trick: (i) prepare $|00\rangle$, (ii) apply $U_\phi(x_j;\beta)$, (iii) apply $U_\phi(x_i;\beta)^\dagger$, (iv) measure; the probability of $|00\rangle$ equals $K(x_i, x_j)$. Construct the 4×4 Gram matrix by estimating each entry with S shots; report S and symmetry $K \approx K^\top$.

(d) Train the SVM with quantum kernel (QSVM) — sweep over C

Solve the dual SVM with kernel K:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N} \alpha_i - \tfrac{1}{2}\sum_{i,j} \alpha_i\alpha_j\, y_i y_j\, K(x_i, x_j) \quad \text{s.t.} \quad 0 \le \alpha_i \le C,\ \sum_i \alpha_i y_i = 0.$$

Run this for a grid of soft–margin values $C \in \{0.1,\ 1,\ 10\}$. For each C:

(i) Recover the bias term b from any support vector x_s (and determine all such x_s) with $0 < \alpha_s < C$:

$$b_s \;=\; y_s \;-\; \sum_{i=1}^{N} \alpha_i\, y_i\, K(x_i, x_s).$$

Explanation: x_s is a training input whose Lagrange multiplier α_s lies strictly between the bounds (a true margin support vector). Its label is $y_s \in \{-1, +1\}$. Using such a point enforces the margin condition exactly, yielding an unbiased intercept estimate b_s.

Practical recipe (explicit formulas). Define a small numerical tolerance $\tau > 0$ (e.g., $\tau = 10^{-6}$) and the set of margin support vectors

$$S_0 \;=\; \big\{\, s \in \{1, \ldots, N\} \;:\; \tau < \alpha_s < C - \tau \,\big\}.$$

For each $s \in S_0$, compute

$$b_s \;=\; y_s \;-\; \sum_{i=1}^{N} \alpha_i\, y_i\, K(x_i, x_s).$$

Then average to obtain the final bias:

$$b \;=\; \frac{1}{|S_0|} \sum_{s\in S_0} b_s.$$

Edge case (no strict margin SVs). If $S_0 = \varnothing$, use a looser "near-margin" set with $\delta > 0$ (e.g., $\delta = 10^{-3}$):

$$S_{\text{near}} \;=\; \big\{\, s : \min(\alpha_s,\ C - \alpha_s) \;\le\; \delta \,\big\}, \qquad b \;=\; \frac{1}{|S_{\text{near}}|} \sum_{s\in S_{\text{near}}} \Big(y_s - \sum_{i=1}^{N} \alpha_i y_i K(x_i, x_s) \Big).$$

(Equivalently, you may solve a small least-squares fit for b from the KKT margin equations on S_{near}.)

Decision function (for completeness). With the final b in hand, classify by

$$f(x) \;=\; \operatorname{sign}\!\Big(\sum_{i=1}^{N} \alpha_i\, y_i\, K(x_i, x) \;+\; b \Big),$$

and assign

$$\hat{y} = \begin{cases} +1, & f(x) > 0, \\ \\ -1, & f(x) < 0, \end{cases}$$

where $\hat{y}$ is the final decision of the model. Remember $|f(x)|$ determines how certain the model is in term of the classification $\hat{y}$. If $f(x)$ is large then the model is certain of it's classification given the data. If small, there could be a false positive classification (the new data point is actually a member of the other classification class). Note: For a general N class classification problem - all these ideas can readily be generalizable - we just start with the basics though here team to get the idea ☺.

(ii) **Ablation on β.** Study $\beta = 0$ (no entangling term). What happens to K and accuracy? Explain why entanglement ($\beta \neq 0$) is critical.

(iii) **(Optional) QSVM on IBM NISQ with shot sweep.** Optionally estimate K on real hardware with $S \in \{1, 10, 100\}$ shots per (i, j) and compare to a statevector (no shots) and a shot-based simulator. Report observations.

(iv) **Does C have impact?.** Report for each C value: training accuracy on the four XOR points (e.g. did the point get classified right or not). - and a table of f(x) values for each input (is the model uncertain in its classification?). Did C have impact?

Bonus B — Theoretical Aspects of Variational Quantum Circuits (25 pts)

Summary. This bonus explores the theory of VQCs and the barren plateau effect. You will (i) derive the parameter-shift rule for one component of the gradient, (ii) show that the expected gradient norm decays as $\mathcal{O}(1/2^n)$, and (iii) apply this to state preparation to estimate the prefactor and the qubit threshold where gradients vanish.

The cost function is

$$C(\boldsymbol{\theta}) = \langle 0|^{\otimes n} U^{\dagger}(\boldsymbol{\theta}) \, H \, U(\boldsymbol{\theta}) \, |0\rangle^{\otimes n},$$

with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k) \in \mathbb{R}^k$ and gradient

$$\nabla_{\boldsymbol{\theta}} C = \left(\frac{\partial C}{\partial \theta_1}, \ldots, \frac{\partial C}{\partial \theta_k} \right)^{\top}.$$

(a) **Parameter-shift rule for one component.** For $U(\theta) = e^{-i\theta P/2}$ with Hermitian P and $P^2 = I$, show

$$\frac{\partial C(\theta)}{\partial \theta} = \frac{1}{2}\Big[C\big(\theta + \tfrac{\pi}{2}\big) - C\big(\theta - \tfrac{\pi}{2}\big) \Big].$$

Clarification: this is *one component* of $\nabla_{\boldsymbol{\theta}} C$. For a k-parameter circuit,

$$\frac{\partial C}{\partial \theta_i} = \frac{1}{2}\Big[C(\boldsymbol{\theta} + \tfrac{\pi}{2}\mathbf{e}_i) - C(\boldsymbol{\theta} - \tfrac{\pi}{2}\mathbf{e}_i) \Big].$$

Here, $\mathbf{e}_i$ denotes the standard basis vector in parameter space $\mathbb{R}^k$, i.e.,

$$\mathbf{e}_i = (0, 0, \ldots, 1, \ldots, 0)^{\top},$$

with a single 1 in the i-th position and zeros elsewhere. It isolates the i-th parameter so that the shift applies only to θ_i while keeping all other parameters fixed.

(b) **Expected gradient magnitude (barren plateaus).** For random parameters on n qubits, show:

$$\mathbb{E}\left[\|\nabla_{\boldsymbol{\theta}} C\|^2\right] = \mathcal{O}\left(2^{-n}\right).$$

(Hint: Haar-averaging yields $\mathbb{E}[|\langle 0|U^\dagger O U|0\rangle|^2] = \mathrm{Tr}(O^2)/2^n$.)

(c) **Explicit constant and state-preparation case.** Let

$$C(\boldsymbol{\theta}) = 1 - |\langle \psi_{\text{target}}|U(\boldsymbol{\theta})|0\rangle|^2,$$

with $\mathbb{E}[(\partial_\theta C)^2] \approx C_0/2^n$.

(i) For $C_0 \approx \frac{1}{4}$, solve $C_0/2^{n_{\max}} \geq 10^{-3}$ for $n_{\max}$.

(ii) Interpret the implication for gradient-based VQE/VQC scalability.

(iii) Discuss why C_0 matters (depth/parameter correlations can change C_0 but not the exponential scaling).