Quantum Machine Learning Seminars

Week 1: Foundations of Quantum Mechanics for Quantum Computing

Dr. Hassan Alshal

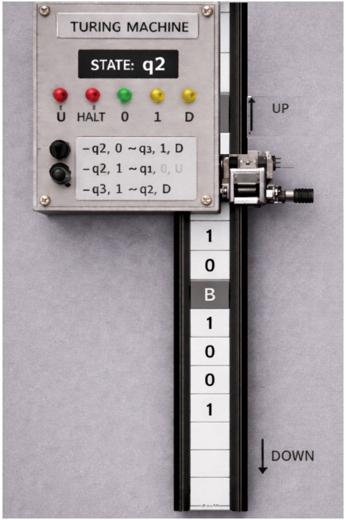
Outline

1. Classical Computability Theory
2. Linear Algebra

Classical Computability Theory

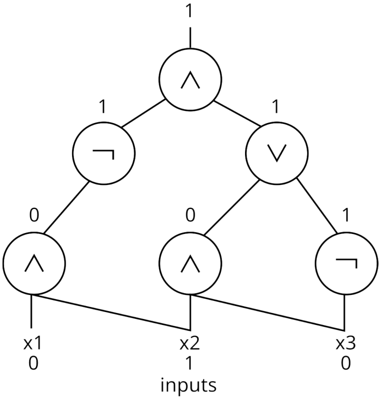
What is computability?

- It's a branch of mathematical logic that studies the limits of what can be computed by algorithms; it classifies problems as solvable (computable) or unsolvable (uncomputable) by theoretical models.
- An algorithm is a step-by-step procedure (the order matters) or rules to accomplish a specific task or solve a problem.
- Theoretical computational models comprehend how the output is obtained from the input, e.g., Turing machine, RAM, decision tree, logic gates, recursive function ... etc.



Limitations of Classical Computability

- Polynomial-time or Polynomial-sized circuit: $T(n) = \mathcal{O}(n^k)$ where $k \neq n$ or its multiples.
- $\boxed{\text{Shannon}}$ $f : \{0,1\}^n \to \{0,1\}$ needs $T(n) \geq \mathcal{O}(2^n/n)$.
- A decision problem that can be solved "efficiently" by a deterministic Turing machine is in class $\mathcal{P}$.
- A decision problem that can be solved by a non-deterministic Turing machine is in class $\mathcal{NP}$.
- Non-deterministic does NOT mean probabilistic, i.e. $\mathcal{BPP} \subset \mathcal{NP}$
- $$\boxed{\mathcal{P} \subset \mathcal{NP} \quad \text{but} \quad \mathcal{P} \neq \mathcal{NP}??}$$

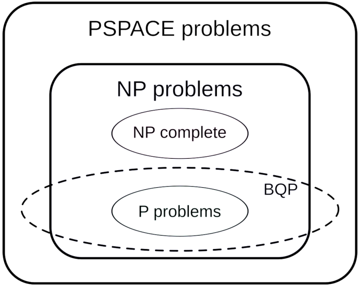


https://en.wikipedia.org/wiki/Boolean_circuit

Classical Computability Theory

$\mathcal{BQP}$

- It has an error probability of at most 1/3 for all instances.
- They explore many possibilities at the same time using superposition.
- The outcome has a tiny chance of error as interference enforces correct paths and cancels wrong ones.
- Solving integer factorization in polynomial time is a $\mathcal{BQP}$ not $\mathcal{BPP}$.



https://en.wikipedia.org/wiki/BQP

Summary

- Vector addition:

$$(\vec{v}+\vec{w})_j = v_j + w_j$$

- Vector scalar multiplication:

$$(c\vec{v})_j = cv_j$$

- Matrix addition:

$$(M+N)_{ij} = M_{ij} + N_{ij}$$

- Matrix scalar multiplication:

$$(cM)_{ij} = c(M_{ij})$$

Summary

- Matrix multiplication:

$$(MN)_{ij} = \sum_k M_{ik} N_{kj}$$

- Matrix/vector complex conjugate:

$$(M^*)_{ij} = (M_{ij})^*$$

- Matrix/vector transpose:

$$(M^T)_{ij} = M_{ji}$$

- Matrix/vector conjugate transposed:

$$(M^\dagger)_{ij} = ((M^*)^T)_{ij} = (M_{ji})^*$$

Linear Algebra

Summary

- Inner / Scalar / Dot product:

$$\vec{v} \bullet \vec{w} = \vec{v}^{\dagger}\vec{w} = \sum_{j=1}^{n} v_j^{T*} w_j$$

- Norm of a vector:

$$\|\vec{v}\| = \sqrt{\vec{v} \bullet \vec{v}}$$

- Projection of $\vec{v}$ onto $\vec{w}$:

$$P_{\vec{v},\vec{w}} = \frac{1}{\|\vec{w}\|}\vec{w} \bullet \vec{v}$$

Linear Algebra

Summary

- Eigenvectors and eigenvalues:

$$M\vec{v} = \lambda\vec{v}$$

$$[M - \lambda\mathbb{I}]\vec{v} = \vec{0}$$

$$\det[M - \lambda\mathbb{I}] = 0$$

- Matrix diagonaliztion

$$P^{-1}MP = D$$

where $P = [v_1 \ v_2 \ \cdots]$ for linearly independent v_i.

Thank You!