Quantum ML

Week 4 - 2019 AI Inspire

https://www.youtube.com/watch?v=hWx2Rws3L-A

https://www.youtube.com/watch?v=7MPcq8z8jbo

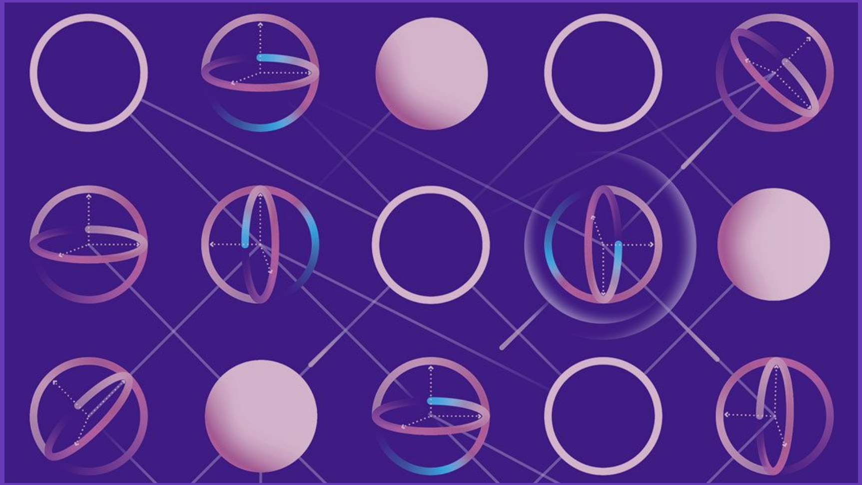
Concepts

Recap of Machine Learning

- Supervised - Input = Labeled training data input → map input to output
- Unsupervised - Input = Unlabeled training data
- Semi-supervised - Input = Small amt of labeled data and large amt of unlabeled
- Reinforcement Learning - Learn to take best course of action to maximize reward in particular environment

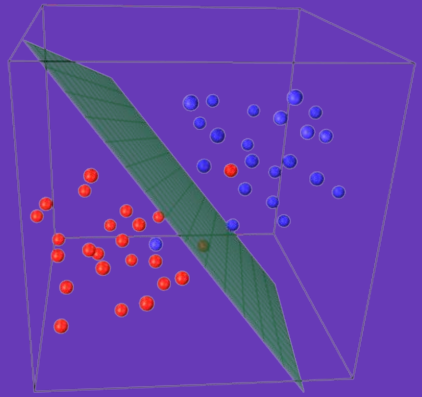
Making the Leap from ML → Quantum ML

- ➢ Classical Machine Learning + Quantum Computing
- ➢ Quantum computing → harnesses quantum physics concepts and quantum circuits + qubits & Linear Algebra

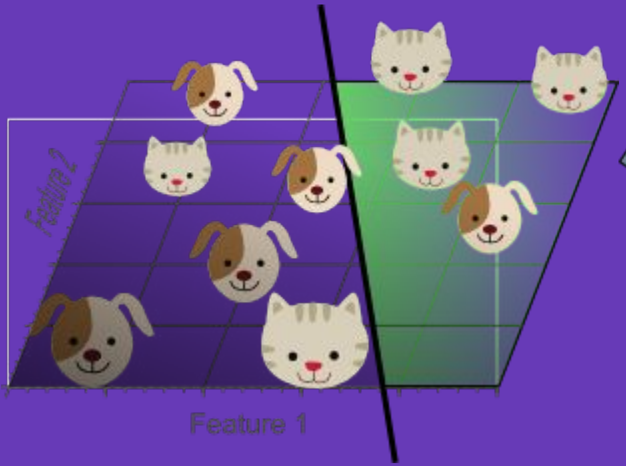


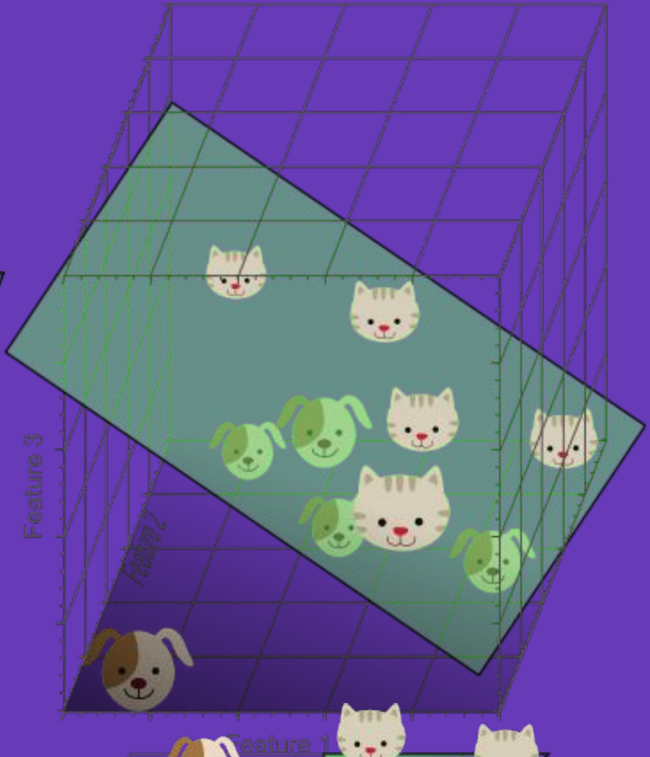
What is It?

- Intersection of Quantum Computing and Machine Learning
- Used for nth dimensions (harder to do on classical computer)
- Can classify between multiple diff classes
- Classical computers → easiest for 2 dimensions



What is It?



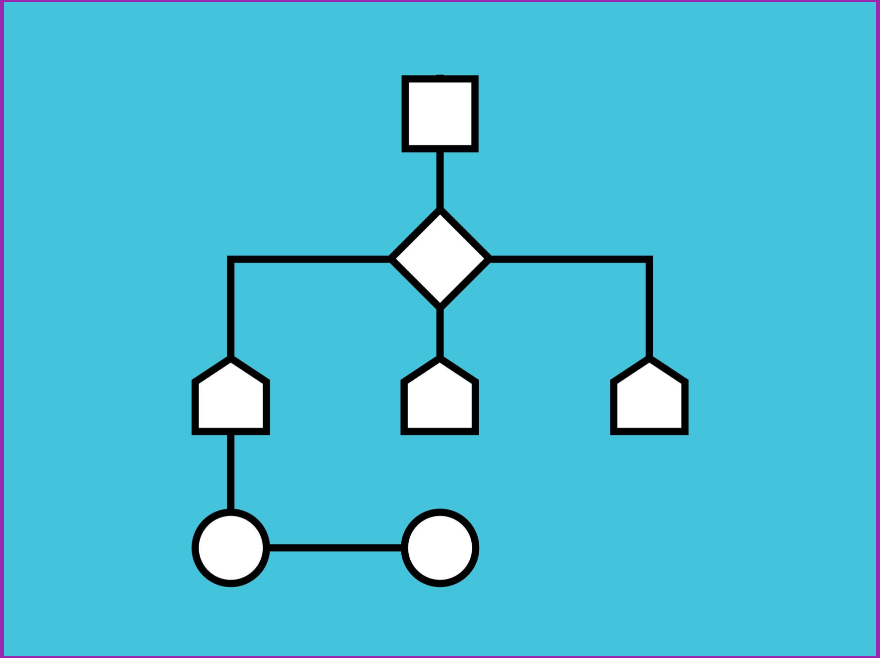


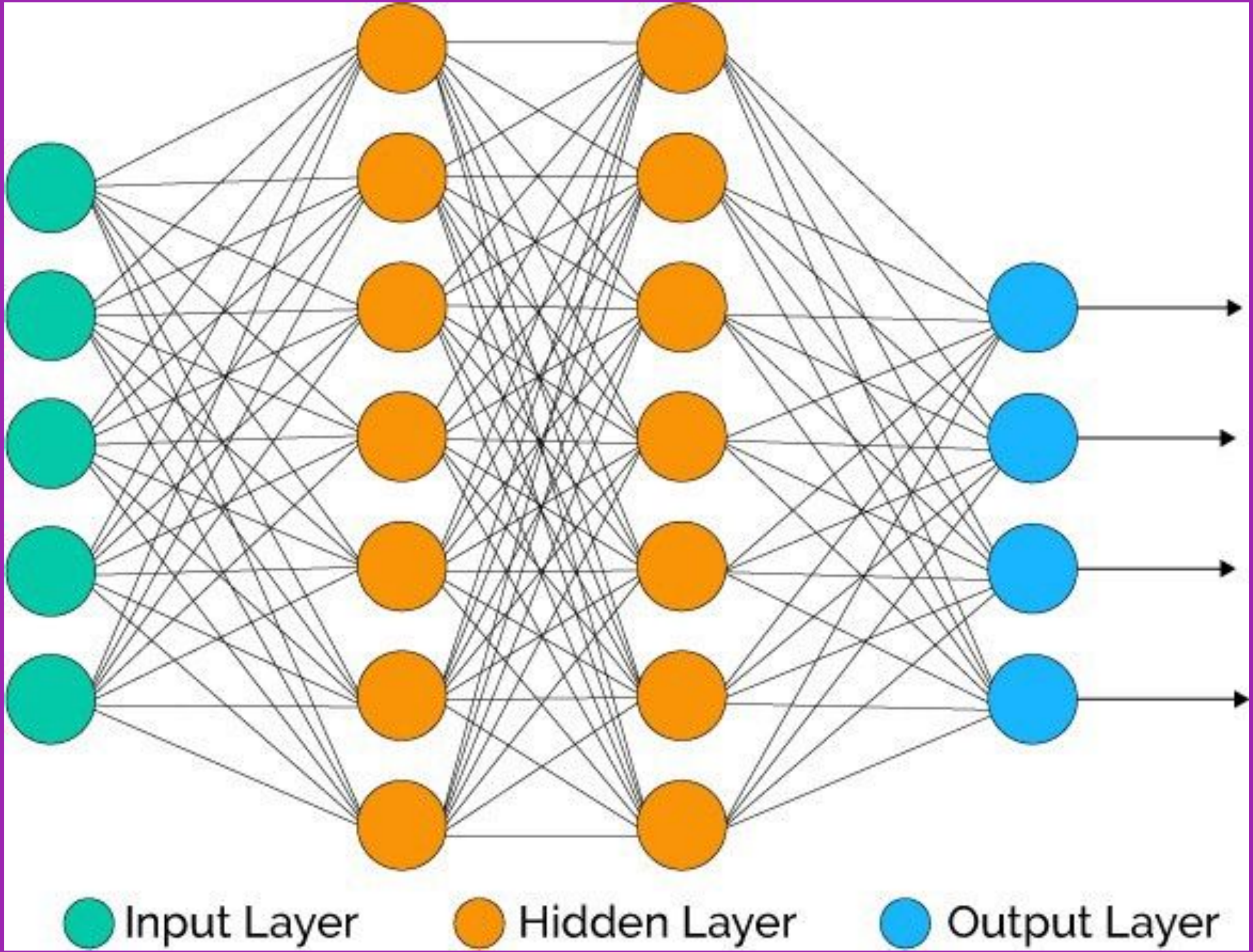
Try to distinguish cats from dogs based on features
Ex - Feature 1 - blue color in dogs and cats

Feature 2 - amt of green color

Feature 3 - amt of red color

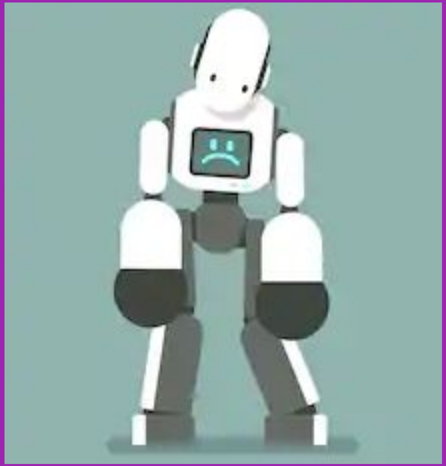
Algorithms




Input Layer
Hidden Layer
Output Layer

Challenge of Machine Learning Neural Net

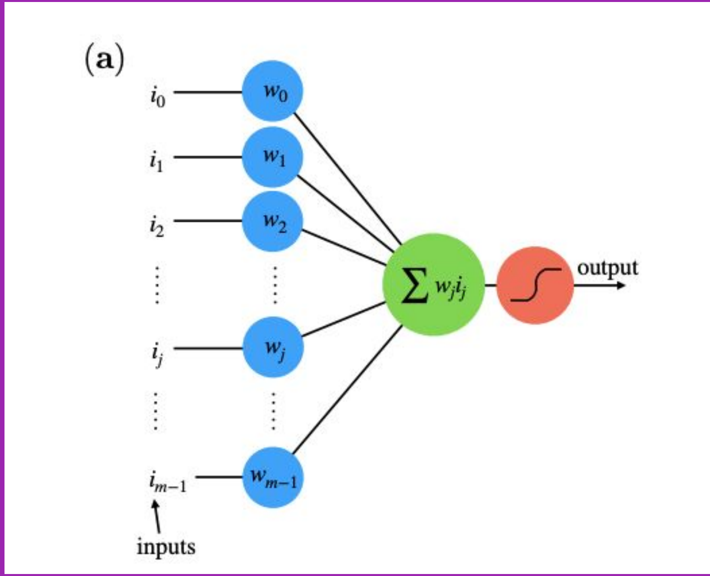
- A lot of data required to train neural network
- Lot of data → lot of time to make decisions
- Weeks to months amt of time
- Can accelerate training process using Quantum ML

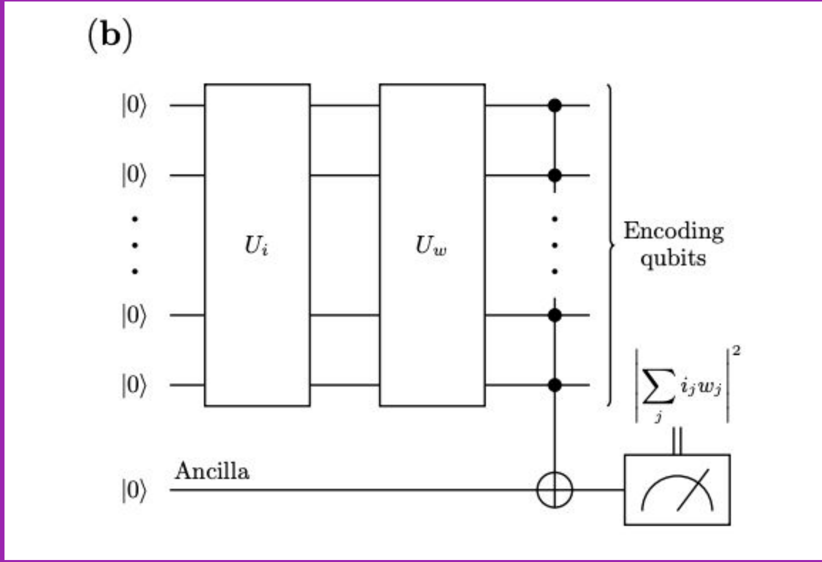


Quantum Neural Network (QNN)

- ➢ Classical neural nets use 1 network → identify patterns
- ➢ QNN uses several networks → identify patterns
- ➢ Uses quantum physics to use several networks
  - ○ Superposition - possible for qubit to be in many states @ same time
- ➢ QNN → stores all possible patterns in a qubit simultaneously

CNN vs QNN





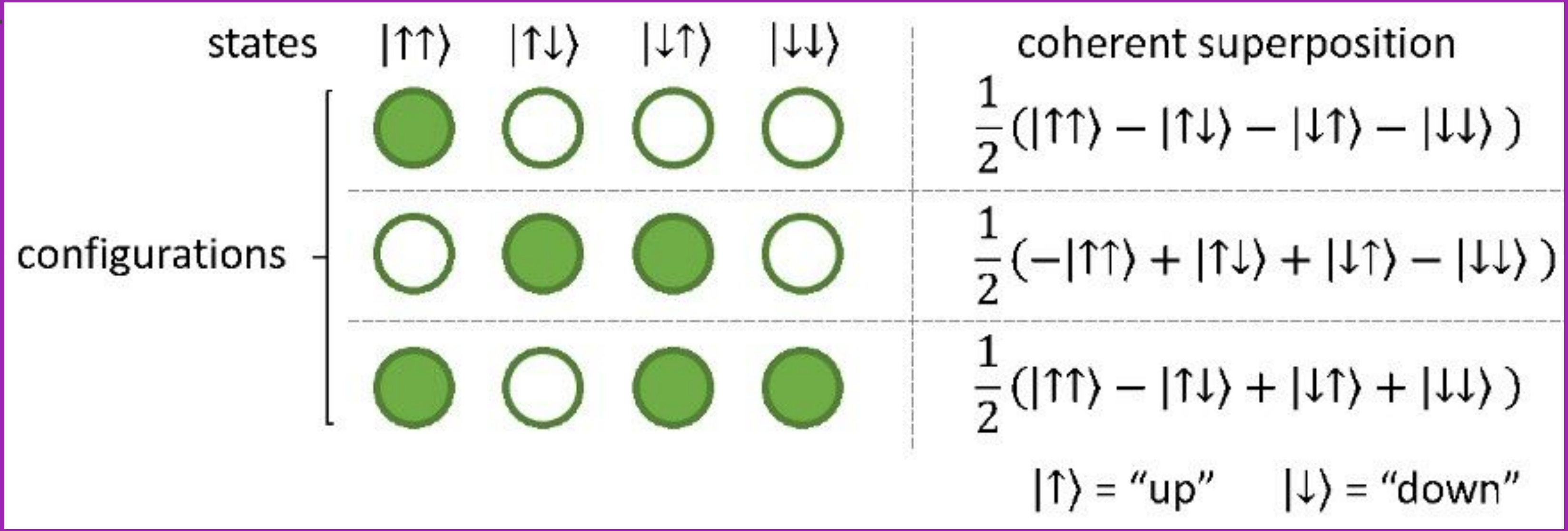
Output = weighted sum of input vector which passes through sigmoid activation function

1st layer: encodes input vector → quantum states
2nd layer : unitary transformations on input (weight vector)
Output : Ancilla qubit (stores entangled state)

QNN Advantages

- Large memory capacity
  - Exponential
- Faster learning
- More stable + reliable
- Able to process much faster

QNN

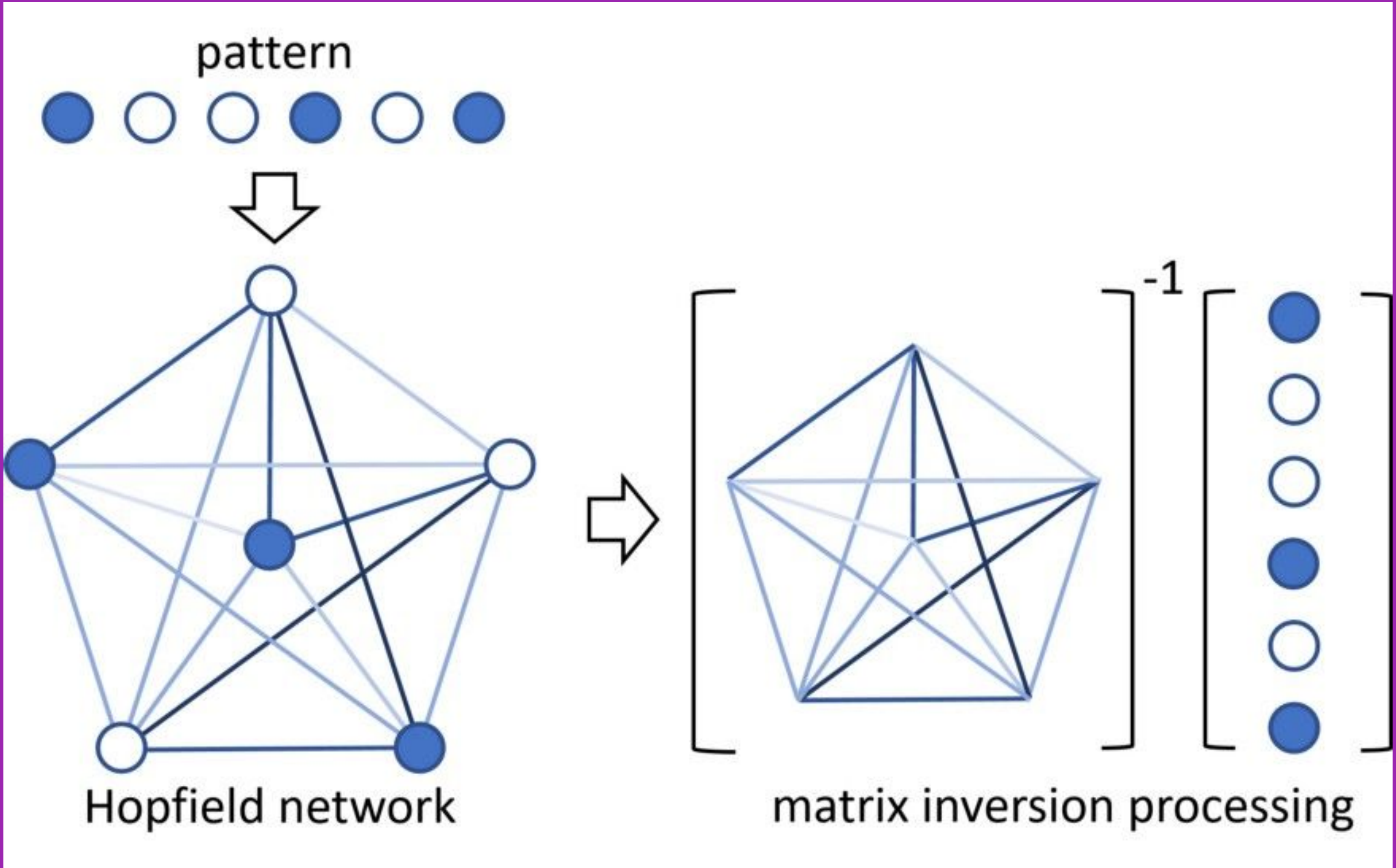


Embedding neurons within qubits
Choosing good neural net is important → performance
STORING ALL POSSIBLE PATTERNS IN QUBITS SIMULTANEOUSLY!!!!

QNN of 4 neurons and 2 qubits
Each qubit is represented by a state (up spin or down spin)
Each neuron → associated with state of system
on neuron = + sign in superposition, off = - sign
System existing in combination of states

QNN Using Hopfield Network

- Hopfield network
  - Type of recurrent neural network
  - All neurons are connected to each other
    - Each node can't be connected to itself
  - Can change weights between different neurons
  - Different configurations of neurons with altering diff weights → diff patterns
  - Objective - Hopfield network has diff patterns in memory & want to load new patterns in memory which are most similar to current ones
- How to update network (update weights so output becomes closer to actual output) - **Classical Method**
  - Continuously pick neurons and look at the weights between the other nodes it's connected to
  - Update each neuron
  - Keep on updating till get error to be close to 0, approach finite value asymptotically



pattern
Hopfield network
-1
matrix inversion processing

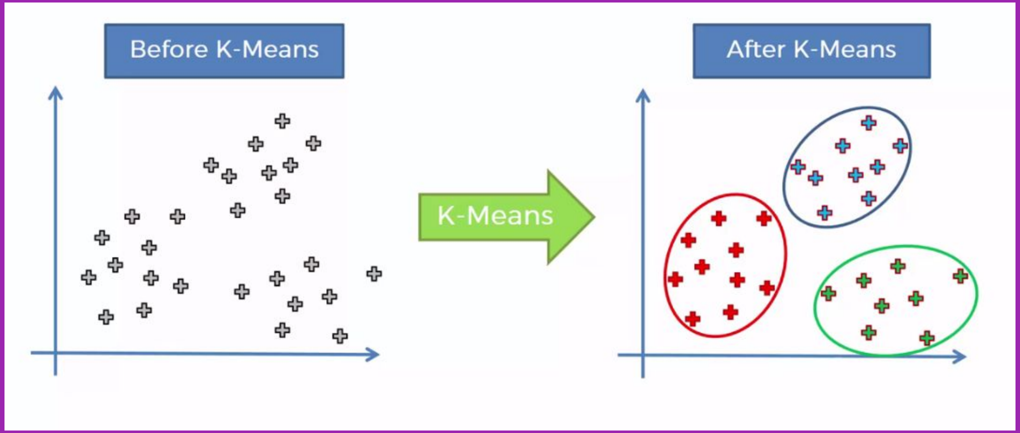
QNN Using Hopfield Network

Procedure -

1. Matrix contains weights between all of the neurons in Hopfield network
   a. Run in 1 step - INVERT matrix (weights needed to come close to actual value happens through inverting matrix)
2. To invert matrix → HHL algo (solve linear system of equations using inverse) → uses Hamiltonian simulation of matrix (quantum Hebbian learning algo)
3. Perform quantum Hebbian learning
   a. Continuously swaps diff qubit banks which have memory patterns
   b. Swapping diff qubits in quantum Hebbian learning → “trains the network”

ML Clustering Algo

- Divide input into k clusters
- Based off of distance measure between centroids of each cluster (initially randomly assigns centroids and gets better value over iterations)
- Classical k-means algo

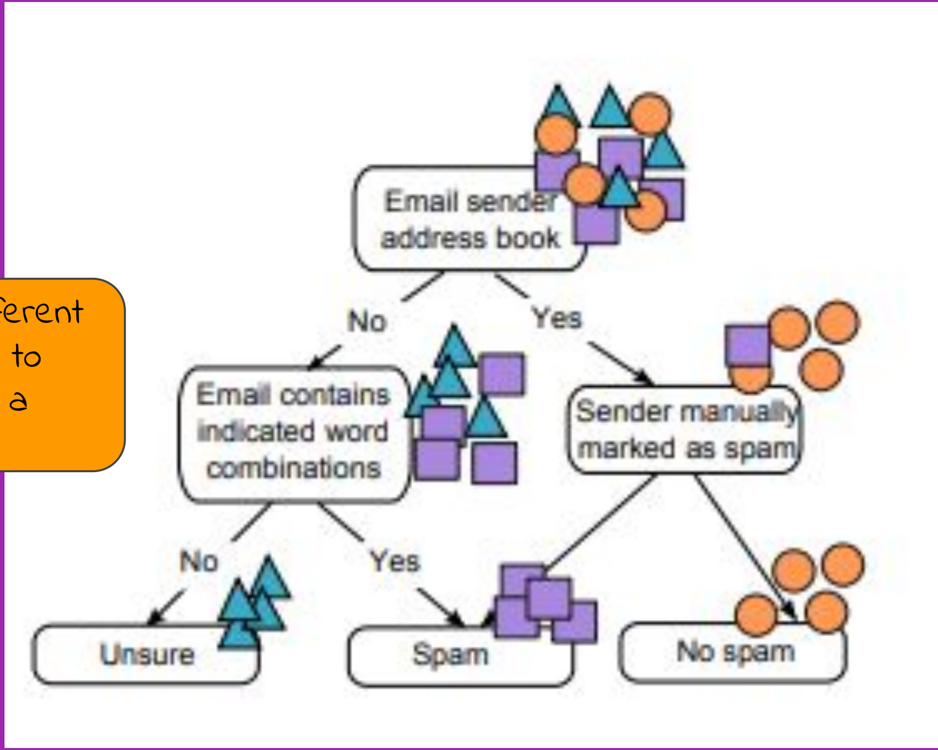


Quantum Clustering Algo

- ➢ Function calculates dist between 2 states
- ➢ For every cluster
  - ○ Calculates summation of distances from each state to other states
  - ○ Quantum state of system can be in quantum superposition → existing in several states @ once
    - ■ Represented as vector → linear algebra needed to compute distances
- ➢ Finds the minimum summation
  - ○ Minimum summation = median for cluster

ML Decision Tree

Split data based on different characteristics/features to reach a consensus with a question



ML Decision Tree Entropy

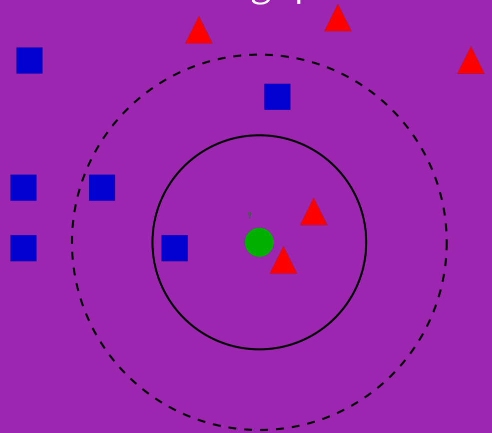
- Entropy = amt of disorder
- Most entropy/disorder in beginning of decision tree
- As you go further down tree and more branches form → less entropy
    - Classes of objects becoming more defined
    - Easier to classify
    - Less entropy/disorder
- Objective - decrease entropy after each cut → if not decreasing, need to split from diff feature/characteristic or stop branch
    - Continuously partitioning/splitting data
    - Least # of branches
    - Start off with broader questions → more specific
- Decision trees use Shannon entropy

Quantum Decision Tree Algo

- Von Neumann Entropy
  - Quantum Mechanics concept for Quantum ML decision trees
  - Parallel to Shannon’s entropy used for Classical ML trees
- Quantum decision tree has not actually been constructed yet
- Algorithm still in theory
  - New type of entropy is similar to entropy used in decision tree
  - Harnesses quantum mechanics

ML K Nearest Neighbors

- ➢ Given input, first identify what k is
- ➢ Start from 1st training point and calculate distances from one point to others
  - ○ Sort these distances
  - ○ Identify which distances are the least from the particular training point
- ➢ Iterate this process for all training points

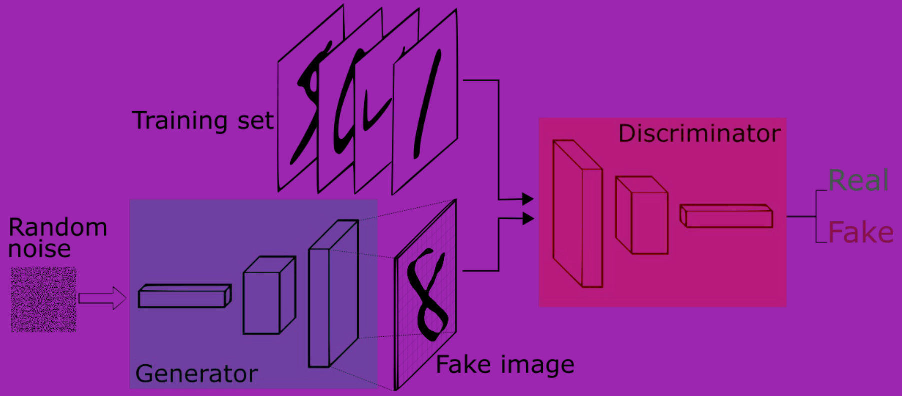


Quantum K Nearest Neighbors Algo

- Objective - calculate distance between quantum states
  - Once distance is calculated, neighbors can be easily determined
- Classical Euclidean distance → Distance between quantum states (rep as vectors)
- Quantum Swap Test
  - Finds the similarity measure between 2 quantum states using Hadamard logic gate in circuit which puts state in superposition (prob that one state can be transferred to other state after imposing superposition = overlap)
- Overlap of 2 states
  - Shows how close 2 states are to each other
    - Probability 1 state will be exactly like other state
      - 0 = no overlap, 1 = full overlap
  - Obtain overlap of 2 states through quantum swap test
- Obtain distance between 2 states from quantum swap state and fidelity/overlap

ML Generative Adversarial Network (GAN)

- ➢ Unsupervised model using supervised loss function
- ➢ 2 competing neural networks
    - ○ Generative and Discriminator
- ➢ Great model which solves issue of not having enough data in real world cases → generates own data and gets trained from that
    - ○ Still several challenges of GAN



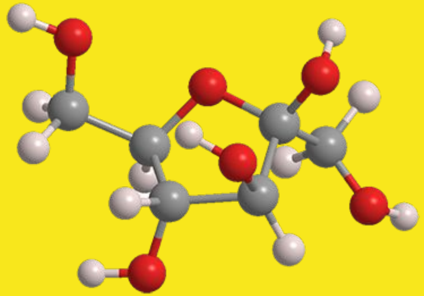
ML Generative Adversarial Network (GAN)

- Generator neural network creates fake images from input vector
- Discriminator network learns to distinguish between the real and false images (real images taken from dataset)
  - Discriminator returns diff prob of each image being real (1 = real and 0 = fake)
- Cop-counterfeiter analogy
  - Counterfeiter learning to pass fake notes and cop is also in training as is able to start distinguishing

Applications

One Application of Quantum ML

- ➢ Classify atoms
- ➢ Hard to simulate on classical computers
- ➢ More atoms added → molecule becomes more complex
  - ○ Number of interactions between atoms in molecule grow very fast
- ➢ Can help with designing new chem compounds
- ➢ Discovery of drugs → solve diseases which can’t be cured



Some other application areas

- More powerful pattern recognition
- Much faster classification
  - Higher dimensions, could be too complex for classical computer
- Developing new types of wearable technologies and materials

Python

PennyLane - Python Library

- Mainly for Quantum Machine Learning
- Compatible with several diff ML libraries
- Optimization + ML tools
- Incorporate quantum circuits
- Hybrid computation
    - Enables both classical and quantum worlds to work together seamlessly

XANADU

PENNY LANE