CLASSIQ

QWORLD

QNNs and Advanced Applications

Quantum Software Development Journey: From Theory to Application with Classiq - Part 4

Program Overview

Quantum Software Development Journey: From Theory to Application with Classiq

- ➢ Week 1: Introduction to the Classiq Platform & High-Level Functional Design
- ➢ Week 2: Git & GitHub - Software Development Skills
- ➢ Week 3: Introduction to Quantum Machine Learning and VQE
- ➢ **Week 4: QNN and Advanced Applications**

Session Overview

QNN and Advanced Applications

Introduction - 30 min

- ➢ Overview of Session 3
- ➢ Neural Networks (NNs)
- ➢ Quantum Neural Networks (QNNs)
- ➢ Auto-Encoders (AEs)
- ➢ Quantum Auto-Encoders (QAEs)

Hand-On Practice - 60 min

- ➢ QNN and QAE Implementation with Classiq

Let’s practice!

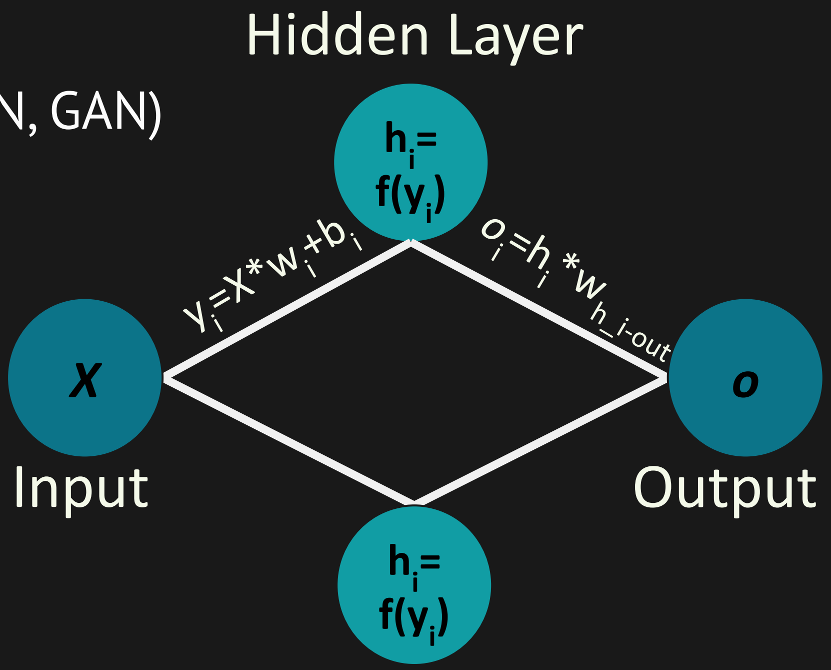
Session 3 - Overview

- ML Key Concepts:
  - Training Data
  - Features (predictors) and Labels (target variable, **in supervised)**
  - Training and Testing Phases
  - Model Evaluation
- Training model:
  - Model Parameters
  - Forward Propagation
  - Calculate Loss
  - Backpropagation and Optimization
- Parametric Quantum Circuits (PQC) & VQE

		Type Algorithm	
		classical	quantum
Type Data	classical	CC	CQ
	quantum	QC	QQ

Neural Networks (NNs)

- NNs are a fundamental component of classical ML
  - Many architectures for different applications (FFN, CNN, RNN, GAN)
  - Composed of neurons (nodes) and connections (edges)
- Forward Pass
- Loss Calculation
- Backpropagation and Optimization
  - Compute gradients
  - Update the network parameters using an optimizer



$\boldsymbol{o}=h_1*w_{h\_1\text{-}out}+h_1*w_{h\_2\text{-}out}+b_o$

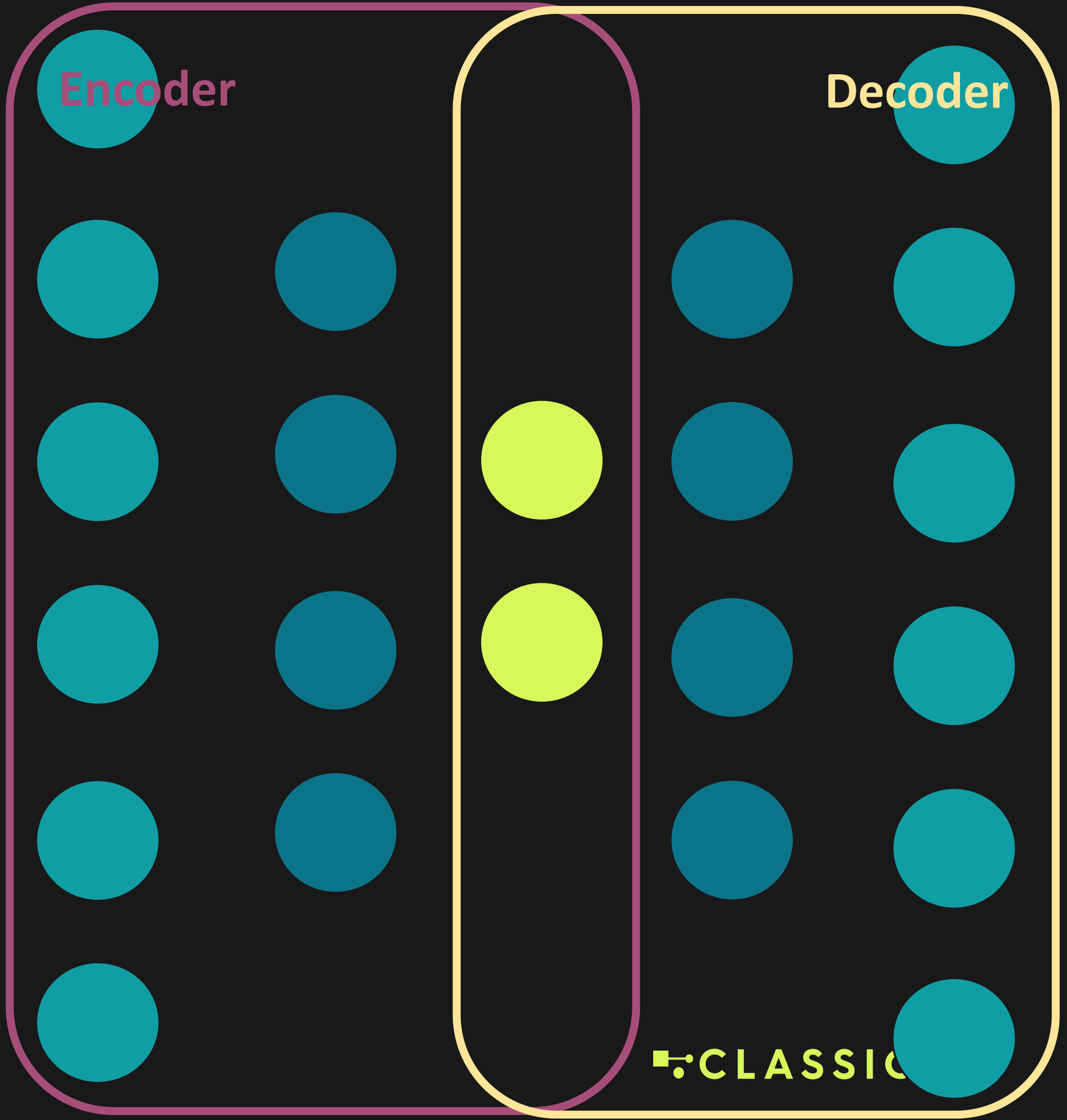
ut

Quantum Neural Networks (QNNs)

- QNNs are models used in Quantum Machine Learning:
  - **Encoding** classical input into quantum state
  - This quantum state is fed into the ansatz
  - The ansatz operates based on its adjustable parameters
    - Could be thoughts of as a “quantum activation function”
  - Measurements and **classical post-processing**
    - The ansatz's output comes from measuring the qubits
    - Classical post-processing is performed, yielding the quantum layer’s output
    - The output can be used as input for another layer, classical or quantum

Auto-Encoders (AEs)

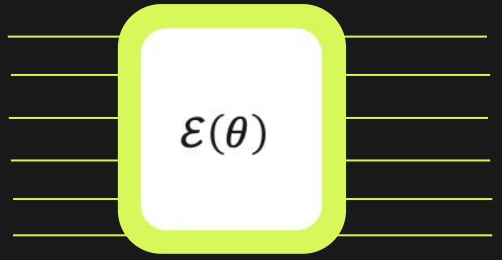
- Autoencoders are neural networks that generalize input data using high-level features
  - Purpose - Dimensionality reduction and data compression
- Components:
  - Encoder - Compresses input data to a lower-dimensional latent space
  - Latent Space - Compact representation of the data
  - Decoder - Reconstructs input data from the latent space



Quantum Auto-Encoders (QAEs)

- QAEs are type of QNNs which are the analogy on AEs, but with quantum layer(s)
  - Purpose - Quantum data compression, anomaly detection and noise reduction
- Could be used on both classical and quantum data
- Quantum Mechanics is reversible and thus:
  - Qubits cannot be "erased", so: **$|Psi>_{n}=|Psi'>_{n-m}|0>_{m}$, $m<n$**
  - The decoder is the inverse unitary operation of the encoder

Quantum Auto-Encoders (QAEs)



data on n qubits

m qubits

$|0\rangle$
$|0\rangle$ $n - m$ qubits at state 0.

Encoder = Parametric circuit

Session Overview

QNN and Advanced Applications

Introduction - 30 min

- ➢ Some Overview of last session
- ➢ Neural Networks (NNs)
- ➢ Quantum Neural Networks (QNNs)
- ➢ Auto-Encoders (AEs)
- ➢ Quantum Auto-Encoders (QAEs)

Hand-On Practice - 60 min

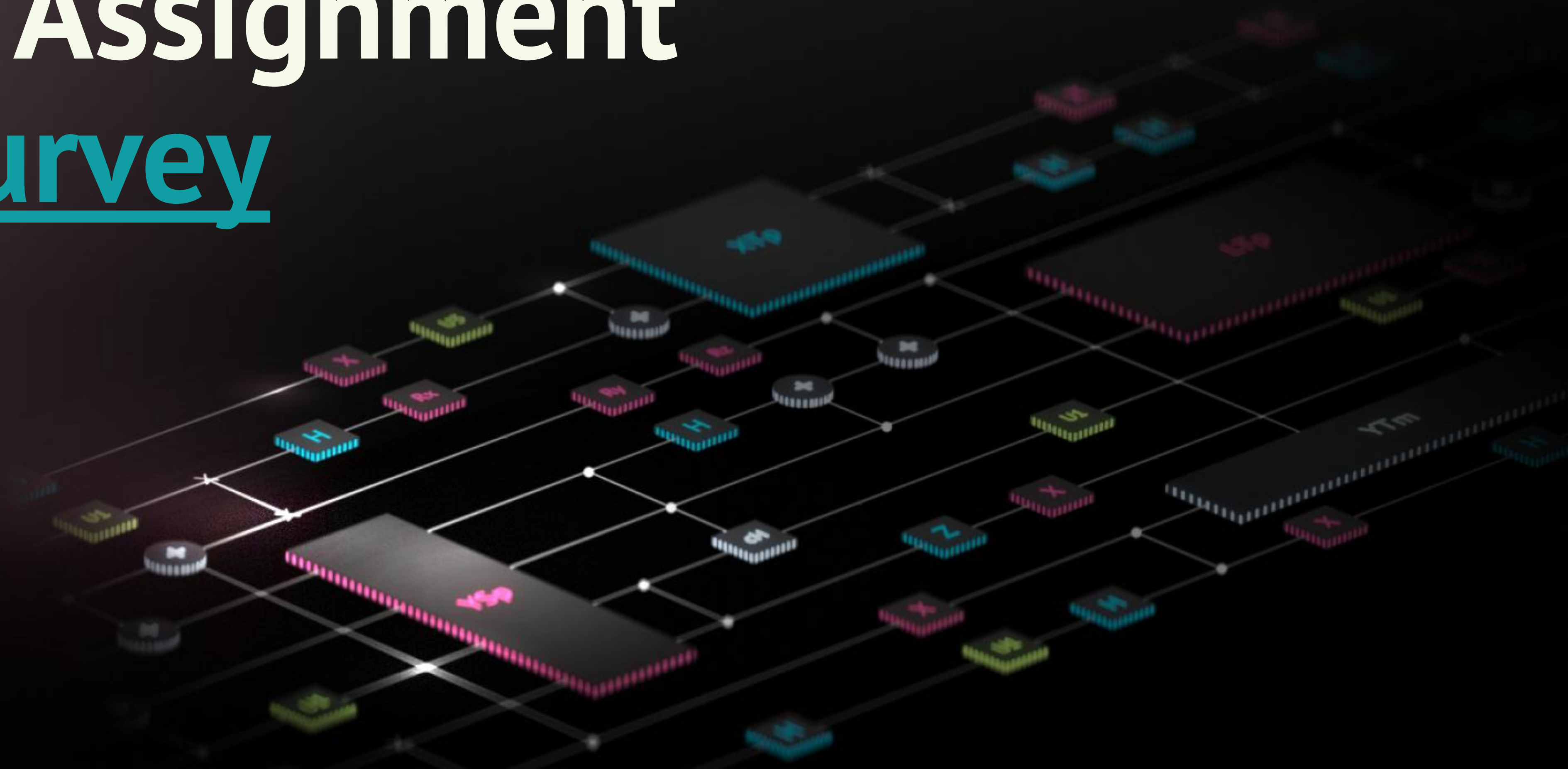
- ➢ **QNN and QAE Implementation with Classiq**

Let’s practice!

CLASSIQ

Last Assignment & Survey

CLASSIQ.IO



CLASSIQ

THANK YOU

CLASSIQ.IO