Womanium Quantum+AI 2024
Quantum-AI-for-Climate
Team Q-MARS

Abdullah Kazi, Maham Khalid , Rishikesh Gokale & Sara Iatrache

.

a) Problem Statement and Background

As Dr. Youzuo Lin succinctly put it: ** "Seismic waves are currently the only effective tool that can penetrate the entire Earth, and seismic inversion (tomography) is used to obtain the structural information of the Earth" ** [1]. This structural information has led to the discovery of oil wells and the monitoring of geothermal energy extracted from the Earth's crust, along with other subsurface features. By accurately mapping these structures Seismic inversion is a straightforward yet powerful technique. It involves transforming routine reflectivity data, which typically highlights interfaces like geological boundaries, into rock properties known as impedance—calculated by multiplying sonic velocity and bulk density. In conventional seismic sections, high-amplitude reflections mark geological formation boundaries, such as the top of a reservoir, making them well-suited for structural analysis. In contrast, inverted data provides insights into internal rock properties, such as lithology, porosity, and fluid types (like brine or hydrocarbons). This makes seismic inversion particularly valuable for stratigraphic interpretation and reservoir characterization.

a) Problem Statement and Background

.

Seismic impedance reveals extensive geological information about the subsurface. Accurate seismic impedance inversion is essential for interpreting seismic data effectively [2]. This process can be divided into two categories: linear impedance inversion and nonlinear impedance inversion. Linear impedance inversion assumes a direct, proportional relationship between seismic data and impedance. However, nonlinear impedance inversion accounts for more complex, non-proportional relationships by employing nonlinear models and algorithms, capturing intricate geological variations.
The two major challenges of seismic inversion are that it is often an ill-posed problem with multiple possible solutions that can fit the data, making it difficult to obtain a unique and accurate result [3]. Additionally, noise in the seismic data can obscure crucial information and degrade the quality of the inversion, further complicating the process.Since Galileo [4], the two fundamental pillars of science have been experimentation and theorization.

a) Problem Statement and Background

Problem Statement: Seismic data processing is crucial for resource exploration and monitoring, including oil and gas exploration and environmental monitoring. Traditional methods of analyzing seismic data can be computationally intensive and may not fully capture the complex patterns present in the data. This project aims to explore and apply advanced quantum methods to enhance the accuracy and efficiency of seismic data analysis.

a) Problem Statement and Background

Importance to the Team: Our team is interested in solving this problem because quantum computing holds the potential to significantly improve the performance of complex data analysis tasks. The unique computational power of quantum algorithms could provide new insights and more accurate models for interpreting seismic data. This could lead to better resource management, more efficient exploration processes, and advancements in environmental monitoring. b) Background Research and Literature Review Background Research: Traditional Methods:

Fourier Transform and Wavelet Transform: Widely used for analyzing seismic signals. They transform the data into frequency domain to identify patterns and anomalies. Machine Learning Models: Techniques such as Convolutional Neural Networks (CNNs) have been applied to seismic data to classify and predict subsurface structures.

Quantum Methods (cont.)

Quantum Convolutional Neural Networks (QCNNs)

QCNNs use quantum gates to perform convolutions and pooling, potentially enhancing the feature extraction process from seismic data. They can capture more complex patterns compared to classical CNNs, though they are currently limited by hardware constraints.

Results and Computational Resources

Results

Preliminary results indicate that quantum models, especially QCNNs and QAEs, offer potential improvements in accuracy and feature extraction compared to classical methods. However, these results are often constrained by the current availability of quantum hardware and computational resources.

Computational Resources

Quantum methods require specialized quantum processors, which are currently limited in availability. For simulations, significant computational resources are needed, including high-performance classical computers or quantum simulators.

Advantages of AI and Quantum

AI

Provides robust methods for pattern recognition and prediction. It is mature and widely accessible but may face limitations in handling very large datasets or complex feature interactions.

Quantum

Offers potential advantages in processing complex patterns and optimizing models beyond classical capabilities. Quantum methods can handle high-dimensional data more efficiently but require advanced hardware and resources.

Developing a Demo App

Objective

Develop a small demo application that showcases the quantum methods applied to seismic data analysis. This could involve integrating quantum algorithms with existing seismic data processing workflows to demonstrate their potential benefits.

Approach

Use Existing Code and Data

Implement the quantum models using existing libraries and datasets. For example, use PennyLane or Qiskit for quantum computing and PyTorch or TensorFlow for integrating classical and quantum models.

Demonstration

Create a demo app that allows users to input seismic data and observe the results of quantum-enhanced analysis. This could include visualization of processed data and comparisons with classical methods.

Challenges

Hardware Limitations

Quantum hardware limitations may restrict the scope of the demo.
Use quantum simulators or hybrid approaches to address this issue.

Advantages of AI

Pattern Recognition

AI, especially through machine learning models like CNNs, excels at recognizing complex patterns in seismic data, improving the accuracy of feature extraction and classification tasks.

Advantages of AI (cont.)

Scalability

AI methods can handle large datasets efficiently and scale to more extensive seismic data analyses. They are adaptable to various seismic data types and structures.

Automated Feature Extraction

AI algorithms can automatically learn and extract relevant features from seismic data, reducing the need for manual feature engineering and potentially uncovering hidden patterns.

Advantages of Quantum Methods

Enhanced Data Processing

Quantum methods leverage quantum superposition and entanglement to process and analyze high-dimensional seismic data more efficiently than classical methods.

Advantages of Quantum Methods (cont.)

Complex Pattern Capture

Quantum algorithms, such as Quantum Autoencoders (QAE) and Quantum Convolutional Neural Networks (QCNN), can capture complex and intricate patterns in seismic data that classical methods might miss.

Optimization Capabilities

Quantum methods, such as Variational Quantum Eigensolver (VQE), offer advanced optimization capabilities, improving the accuracy of seismic data models and enhancing their predictive performance.

Comparative Advantages

AI vs Quantum

AI provides robust, scalable solutions with established methods for pattern recognition and prediction. Quantum methods offer potential breakthroughs in processing efficiency and complex pattern capture but are currently limited by hardware constraints.

Quantum Autoencoder (QAE) for Seismic Impedance Inversion

Architecture

A Quantum Autoencoder (QAE) for seismic data consists of two primary components: the encoder and the decoder.

- **Encoder**: Maps classical data into a quantum state using a sequence of quantum gates, including RY, RX, and CNOT gates. The data is encoded as a quantum state:

  $$|\psi\rangle = U_{\text{encoder}} |\text{data}\rangle$$

  where U_{encoder} denotes the unitary operation applied.
- **Decoder**: Reconstructs the classical data from the quantum state using similar gates as the encoder and measures the expectation values of Pauli-Z operators:

  $$\hat{\text{data}} = \langle\psi|U_{\text{decoder}}^{\dagger} Z U_{\text{decoder}}|\psi\rangle$$

  where U_{decoder} denotes the unitary operation for decoding.

Detailed Architecture of Quantum Autoencoder (QAE)

Detailed Architecture

Encoder:

- **Quantum Gates**: Utilizes a combination of RY (rotation around the Y-axis), RX (rotation around the X-axis), and CNOT (controlled-NOT) gates to transform classical data into a quantum state.
- **Encoding Process**:

$$|\psi\rangle = U_{\text{encoder}} |\text{data}\rangle$$

  where U_{encoder} is the unitary operator composed of these gates.
- **Circuit Design**: The encoder circuit is designed to map classical data vectors to a high-dimensional quantum state, allowing the quantum state to capture intricate patterns in the data.

Detailed Architecture of Quantum Autoencoder (QAE)

Detailed Architecture

Decoder:

- **Reconstruction**: Applies the inverse of the encoding gates to reconstruct the data from the quantum state.
- **Measurement**: The quantum state is measured using Pauli-Z operators:

$$\hat{\text{data}} = \langle\psi|U_{\text{decoder}}^{\dagger} Z U_{\text{decoder}}|\psi\rangle$$

  where U_{decoder} is the unitary operator for decoding.
- **Circuit Design**: The decoder is designed to reverse the transformations applied by the encoder, ideally reconstructing the original classical data.

Quantum Circuit Complexity:

- The complexity of the QAE architecture depends on the number of qubits and gates used. Efficient encoding and decoding require careful optimization of the quantum circuit design.

Mathematical Formulation

The performance is evaluated using:

- **Mean Squared Error (MSE)**:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

  where x_i and $\hat{x}_i$ are the original and reconstructed data points, respectively.

- **Mean Absolute Error (MAE)**:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|$$

Data normalization is performed using Min-Max scaling:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

Data is segmented to fit the quantum circuit with n_{wires}, creating fixed-size chunks.

Results

Results

- **Training**: The QAE model was optimized using the Adam optimizer over 100 iterations.
- **Loss History**: Demonstrates convergence from an initial high loss to a stabilized lower value.
- **Reconstruction Quality**: Plots show the comparison between original and reconstructed seismic traces.
- **Metrics**:
  - MSE: Decreased from 0.271 to 0.003.
  - MAE: Decreased from 0.450 to 0.042.

Results

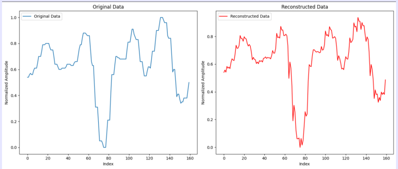


Figure: Original vs Reconstructed

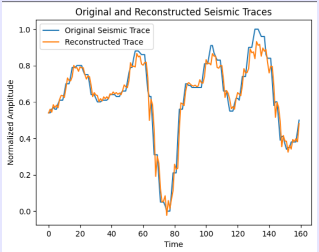


Figure: Original and reconstructed traces

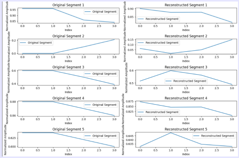


Figure: Original and reconstructed segment

Kernel Principal Component Analysis (PCA) Overview

Kernel PCA Introduction

Objective: Kernel PCA is used to perform dimensionality reduction by mapping the data into a higher-dimensional space where it becomes linearly separable.
Kernel Function: Radial Basis Function (RBF) kernel is utilized:

$$K(x, y) = \exp(-\gamma \|x - y\|^2)$$

where γ is a parameter controlling the spread of the kernel.
Components:

- **Data Scaling**: Data is standardized before applying Kernel PCA.
- **Dimensionality Reduction**: Kernel PCA projects the data onto a lower-dimensional space using the kernel trick.

Data Preparation and Normalization

Data Handling

Seismic Data:

- Loaded from `seismic_trace_15_9_F-15-A.csv`.
- Cropped to specific time ranges and normalized using `MinMaxScaler`.

Statistical Source Wavelet:

- Loaded from `statistical_source_wavelet.txt` for plotting.

Normalization:

- Applied `MinMaxScaler` to seismic trace and impedance trace to ensure values are in the range [0, 1].

Kernel PCA Implementation

Kernel PCA Process

Algorithm Steps:

- **Data Transformation**: Applied Kernel PCA with RBF kernel on standardized data.
- **Dimensionality Reduction**: Reduced data to principal components.

Metrics:

- **Number of Clusters**: 2 clusters identified from the data.
- **Silhouette Score**: Evaluates the quality of clustering; average score: 0.47.

Visualization of Kernel PCA Results

Clustering Visualization

Plots:

- **2D Scatter Plot**: Shows the distribution of data in the first two principal components.
- **Enhanced Circuit Results**: Comparisons with and without quantum-enhanced circuits.

Observations:

- Distinct clustering patterns are observed.
- Enhanced circuit provides a clearer separation in feature space.

Quantum Circuit Architecture

Random Layer Architecture

Random Layers Function:

- `RandomLayers(weights, wires, rotations, seed)`:
  - Applies random layers with specified weights and rotations to qubits.
  - `weights` specifies rotation angles.
  - `wires` indicates qubit wires.
  - `rotations` contains rotation operations like `qml.RX`, `qml.RY`, `qml.RZ`.

Noise Layer Function:

- `noise_layer(angle)`:
  - Applies noise layers with a specified angle.
  - Uses `qml.RZ` and `qml.CNOT` gates.

Circuit Generation and Data Processing

Circuit Generation

Generate Circuit Function:

- `generate_circuit(shots, ts=False)`:
  - Creates a quantum circuit for T-symmetric or general unitary operations.
  - Uses `qml.device("default.qubit")` for simulations.
  - Applies `RandomLayers` with random weights.

Expectation Values:

- Returns expectation values `qml.expval(qml.PauliY(q))`.

Data Processing

Process Data Function:

- `process_data(raw_data)`:
  - Converts raw data to vectors of means and variances for each qubit.
  - Outputs vectors combining means and variances.

Enhanced Quantum Circuit Integration

Quantum Circuit with Noise

Enhanced Circuit Features:

- Includes additional noise layers and Hadamard gates.
- Applied random unitary transformations and noise layer to improve feature extraction.

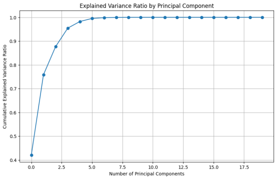
Performance Metrics:

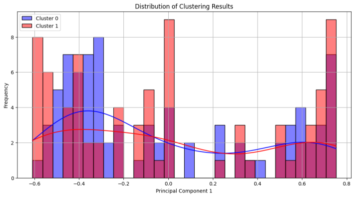
- **Clustering Metrics**: Number of clusters and silhouette score calculated for enhanced circuit.
- **Visualization**: Enhanced circuit results in clearer clustering in principal component space.

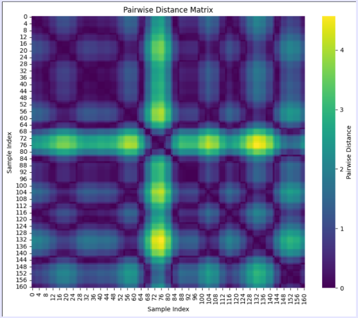
Clustering Results

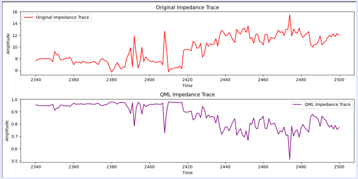
- **Number of clusters:** 2
- **Number of samples:** 60
- **Silhouette Score:** 0.47

- **Number of clusters (enhanced):** 2
- **Number of samples (enhanced):** 60
- **Silhouette Score (enhanced):** 0.21


Explained Variance Ratio by Principal Component
Cumulative Explained Variance Ratio
Number of Principal Components


Distribution of Clustering Results
Cluster 0
Cluster 1
Frequency
Principal Component 1


Pairwise Distance Matrix
Sample Index
Sample Index
Pairwise Distance


Original Impedance Trace
Original Impedance Trace
Amplitude
Time
QML Impedance Trace
QML Impedance Trace
Amplitude
Time

Introduction to Quantum Phase Estimation (QPE)

Quantum Phase Estimation (QPE) is a quantum algorithm used to estimate the phase (or eigenvalue) of a unitary operator. It is a crucial component of many quantum algorithms, including Shor's algorithm for factoring large numbers.

Mathematical Overview:

- Given a unitary operator U and its eigenstate $|\psi\rangle$ with eigenvalue $e^{i\phi}$, QPE estimates the phase ϕ.
- The algorithm uses quantum registers to encode the phase information and applies the Inverse Quantum Fourier Transform (IQFT) to extract the phase.
- The accuracy of the phase estimation depends on the number of qubits used.

QPE Algorithm Architecture

The QPE algorithm can be broken down into the following steps:

1. **Initialization:** Prepare the quantum state. Apply Hadamard gates to create a superposition.
2. **Phase Shift:** Apply controlled phase shifts based on the unitary operator U.
3. **Inverse Quantum Fourier Transform (IQFT):** Apply the IQFT to the phase information encoded in the quantum state.
4. **Measurement:** Measure the quantum state to obtain the estimated phase.

QPE Circuit Components:

- **Hadamard Gates:** Create superposition states.
- **Phase Shift Gates:** Encode the phase information.
- **Controlled Rotations:** Part of IQFT to transform the phase information.

Data Preparation

- **Load Data:** Load seismic data from CSV.
- **Crop Data:** Select data within a specified time range.
- **Normalize Data:** Use Min-Max scaling to normalize the seismic traces and IP traces.
- **Code Snippet:**
  - Load and preprocess seismic data.
  - Normalize seismic traces using MinMaxScaler.

QPE Implementation in Code

QPE Circuit Definition

The QPE circuit is implemented using the following components:

- **Hadamard Gates:** Apply Hadamard gates to all qubits.
- **Phase Shift Gates:** Apply RZ gates for phase shifts.
- **Inverse QFT:** Perform the inverse QFT manually.

- **Cost Function:** Define a cost function for optimization to minimize the negative log of the QPE probabilities.
- **Optimization:** Use AdagradOptimizer to adjust phase angles and minimize the cost function.

Results and Visualization

Visualization

The results include:

- **Seismic Trace with QPE Results:** Plot the original seismic trace and compare it with the QPE results.
- **QPE Probability Distribution:** Plot the probabilities obtained from the QPE circuit.

Post-Processing and Metrics

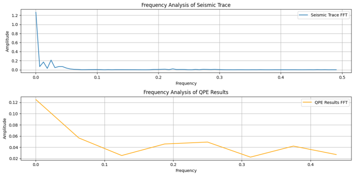
Post-Processing

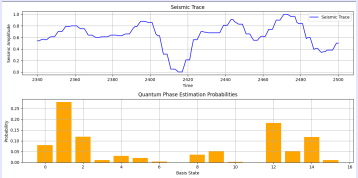
Post-processing includes:

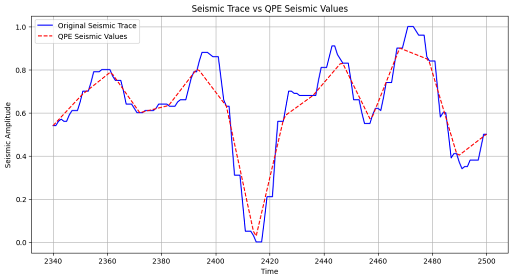
- **Error Histogram:** Compute and plot the histogram of errors between QPE and actual seismic values.
- **Metrics:** Calculate and report Mean Squared Error (MSE) and R-squared (R^2) metrics to evaluate the performance.

- **Error Histogram:** Provides insight into the distribution of errors.
- **Metrics Calculation:** Helps in understanding the fit quality of QPE results to the actual data.

Results


Frequency Analysis of Seismic Trace
Seismic Trace FFT
Amplitude
Frequency
Frequency Analysis of QPE Results
QPE Results FFT
Amplitude
Frequency


Seismic Trace
Seismic Trace
Seismic Amplitude
Time
Quantum Phase Estimation Probabilities
Probability
Basis State


Seismic Trace vs QPE Seismic Values
Original Seismic Trace
QPE Seismic Values
Seismic Amplitude
Time

Hybrid Classical-Quantum Approach for Seismic Data Processing

hybrid classical-quantum approach for seismic data processing using a Variational Quantum Classifier (VQC) to predict acoustic impedance from normalized seismic traces, leveraging quantum computing for potentially improved model performance.

Seismic Trace and Ricker Wavelet

- **Seismic Trace:** A seismic trace records the reflection of seismic waves from subsurface structures. In our context, it has been normalized for further analysis.
- **Ricker Wavelet:** Known for its bell-shaped curve, the Ricker wavelet helps in identifying reflections of subsurface layers.

Ricker Wavelet

The Ricker wavelet $W(t)$ is given by:

$$W(t) = \left(1 - 2\pi^2 f^2 t^2\right) \exp\left(-\pi^2 f^2 t^2\right) \tag{1}$$

where f is the peak frequency and t is the time. The wavelet's shape aids in filtering and analyzing seismic data.

Data Preparation and Normalization

Data Normalization: Normalizes data to a range suitable for model input. We use Min-Max normalization:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{2}$$

where x is the original data, and x_{min} and x_{max} are the minimum and maximum values of the data, respectively.

Variational Quantum Circuit

The quantum circuit is defined by the function `variational_circuit`, parameterized by input data and weights:

$$\text{variational\_circuit}(\text{inputs}, \text{weights}) = \left[\text{qml.expval}(\text{qml.PauliZ}(i)) \mid i \in \{0, \ldots, n_{\text{qubits}} - 1\}\right] \tag{3}$$

- **Angle Embedding:** Encodes classical data into quantum states.
- **Entangling Layers:** Apply parameterized unitary operations to create complex correlations.
- **Observable Measurement:** Measures expectation values of observables (Pauli-Z operators).

Quantum Node

The quantum node is defined using Pennylane's `qml.qnn.TorchLayer`:

$$\text{vqc\_layer} = \text{qml.qnn.TorchLayer}(\text{qml.QNode}(\text{variational\_circuit}, \text{dev}, \text{interface} = "torch", \text{diff\_} \tag{4}$$

where `weight_shapes` defines the shape of the weights:

$$\text{weight\_shapes} = \{"weights" : (5, n_{\text{qubits}}, 3)\} \tag{5}$$

Quantum Neural Network Architecture

- **Quantum Layer:** Encodes data into quantum states and applies variational circuits.
- **Classical Layers:** Traditional neural network layers for processing quantum outputs.
  - **Fully Connected Layer 1:**
    $$h_1 = \text{ReLU}(W_1 \cdot \text{output} + b_1) \tag{6}$$
  - **Fully Connected Layer 2:**
    $$h_2 = \text{ReLU}(W_2 \cdot h_1 + b_2) \tag{7}$$
  - **Output Layer:**
    $$\hat{y} = \text{Sigmoid}(W_3 \cdot h_2 + b_3) \tag{8}$$

Training and Optimization

Loss Function: Mean Squared Error (MSE)

$$\mathsf{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{9}$$

Optimizer: Adam optimizer for adjusting weights based on gradients.
Learning Rate Scheduler: Adjusts learning rate for stable convergence.

Evaluation Metrics

- **Mean Squared Error (MSE):** Measures average squared difference between predicted and actual values.
- **R-squared (R^2) Score:**

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{10}$$

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{11}$$

- **Accuracy:**

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \times 100\% \tag{12}$$

Training Results

The training process for the Variational Quantum Convolutional Neural Network (VQCNN) was conducted over 10 epochs. The training and validation losses for each epoch are summarized as follows:

Epoch	Training Loss	Validation Loss
1	0.0803	0.0872
2	0.0802	0.0871
3	0.0801	0.0870
4	0.0800	0.0868
5	0.0799	0.0867
6	0.0798	0.0866
7	0.0797	0.0864
8	0.0796	0.0863
9	0.0795	0.0862
10	0.0794	0.0860

Table: Training and Validation Losses Over Epochs

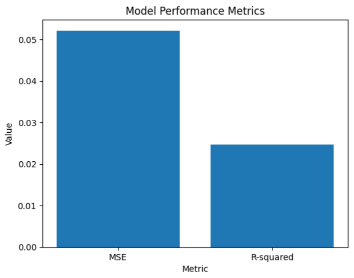
Loss Curves

The plots below illustrate the training and validation losses over the course of training. The final Root Mean Squared Error (RMSE) on the test set is:

$$\text{RMSE} = 9.4092$$

The confusion matrix for the test set is shown below, depicting the performance of the model in binary classification.
The following plots show the seismic trace with added noise and the impedance trace.

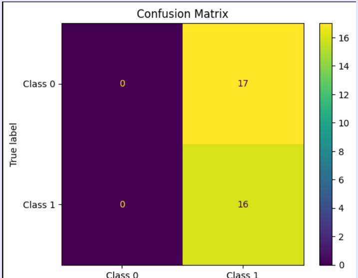
Results



Model Performance Metrics
Value
0.05
0.04
0.03
0.02
0.01
0.00
MSE
R-squared
Metric

Training Loss Over Epochs
Training Loss
Loss
0.0560
0.0555
0.0550
0.0545
0.0540
0.0535
0.0530
0.0525
0
25
50
75
100
125
150
175
200
Epochs

Training and Validation Loss Over Epochs
Training Loss
Validation Loss
Loss
0.087
0.086
0.085
0.084
0.083
0.082
0.081
0.080



Confusion Matrix
True label
Class 0
Class 1
0
17
0
16
16
14
12
10
8
6
4
2
0
Class 0
Class 1

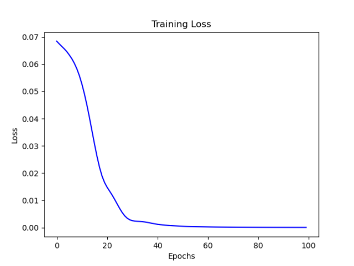
Fully Classical Autoencoder Network

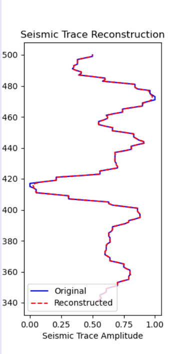
A classical autoencoder is explored for benchmarking purposes. The autoencoder used consists of:

- Total inner layers: 3
- Activation functions: Rectified Linear Unit, Sigmoid
- Loss function: Mean Squared Error
- Optimizer: Adam optimization
- Learning rate: 0.003
- Training Epochs: 100

Results - Classical Autoencoder

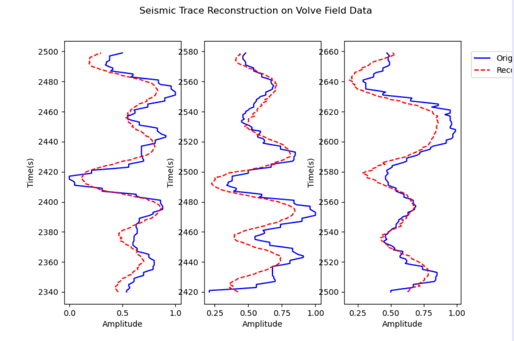
With a single trace as the training data, the seismic trace is reconstructed with nearly 100% accuracy.





Results - Classical Autoencoder

Testing on synthetic data:



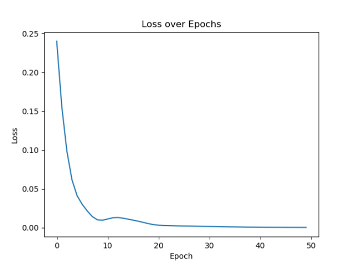
Fully Classical Artificial Neural Network

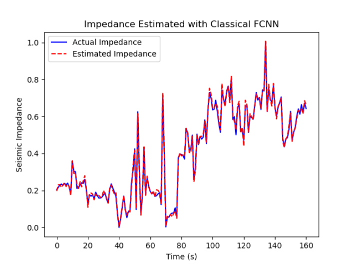
The classical neural network architecture is simple, consisting of:

- Hidden layers: 1
- Activation function: Rectified Linear Unit
- Loss function: Mean Squared Error
- Optimizer: Adam optimization
- Learning rate: 0.002
- Training Epochs: 50

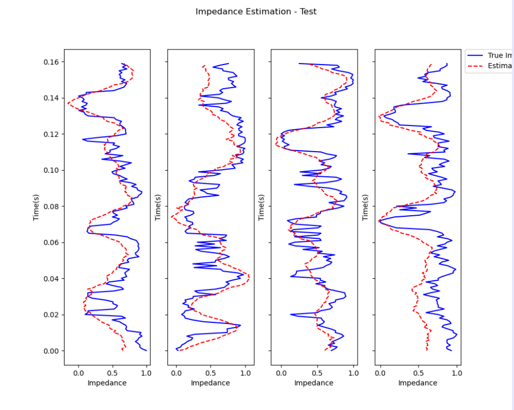
This network is able to accurately predict the seismic impedance given a seismic trace. When extended to more training and testing data, the prediction is able to retain the character of the impedance, though some features are lost.

Results - Classical FCNN





Results - Classical FCNN



Benchmarking Comparison of Quantum Methods

Method	RMSE	MSE	MAE
Quantum Autoencoder (QAE)	9.4092	0.003	0.042
Kernel PCA + Quantum Circuit	-	-	-
Variational Quantum Convolutional Neural Network (VQCNN) (200 Epochs)	-	0.0521	0.2056
Quantum Phase Estimation (QPE)	-	-	-
Classical Autoencoder	-	3.68e-05	0.003
Classical Neural Network	-	1.60e-04	9.98e-03

Table: Benchmarking Comparison of Different Quantum Methods for Seismic Data

References

- Albino, A. et al., 2022. Employing gate-based quantum computing for travel time seismic inversion, vol. 2022(1): 1–5, doi:https://doi.org/10.3997/2214-4609.2022.80007.
- Bravo-Prieto, C. et al., 2019. Variational quantum linear solver, doi:10.48550/ARXIV.1909.05820, URL https://arxiv.org/abs/1909.05820.
- Cerezo, M. et al., 2021. Variational quantum algorithms, Nature Reviews Physics, vol. 3(9): 625–644, doi:10.1038/s42254-021-00348-9.
- Deshpande, A. et al., 2022. Quantum computational advantage via high-dimensional gaussian boson sampling, Science Advances, vol. 8(1), doi:10.1126/sciadv.abi7894.
- Egger, D. J., Marec ek, J. & Woerner, S., 2021. Warm- starting quantum optimization, Quantum, vol. 5: 479, doi:10.22331/q-2021-06-17-479.
- V. Das and A. Pollack. Convolutional neural network for seismic impedance inversion. Geophysics, 84:R869–R880, 2019.

Note: More references for analysis and classical machine learning methods for seismic data are included in further documentation.