A Machine Learning Approach to FFT in OFDM Receiver

May 13th 2024

Orthogonal Frequency Division Multiplexing (OFDM) systems rely heavily on the Fast Fourier Transform (FFT) and its inverse (IFFT) for signal modulation and demodulation. However, while FFT is effective in converting signals between time-domain and frequency-domain representations, it poses computational challenges, particularly in high-speed communication systems. This project aims to address this limitation by replacing the conventional FFT process in OFDM receivers with a machine learning model. Specifically, we propose using feed forward neural networks to perform the FFT operation with similar or improved performance. By leveraging machine learning, we aim to make the FFT process faster while maintaining the accuracy of traditional methods.
Keywords: Prospective PhD Student, OFDM, Research, Deep Learning.

1 Introduction and Background

The Fast Fourier Transform (FFT) and its inverse (IFFT) are fundamental operations in Orthogonal Frequency Division Multiplexing (OFDM) systems. In an OFDM system, the FFT is utilized at the transmitter side to convert time-domain signals into frequency-domain signals. This conversion is essential for efficient modulation and demodulation processes. Initially, a string of data bits is mapped onto complex symbols according to the modulation scheme used, such as Quadrature Amplitude Modulation (QAM) or Phase Shift Keying (PSK). These symbols are then assigned to different subcarriers. The FFT operation takes these modulated symbols, represented in the frequency domain, and transforms them into a time-domain signal. At the receiver side, the Inverse Fast Fourier Transform (IFFT) prepares the received signal for further processing. By converting frequency-domain signals back into time-domain signals, the IFFT enables signal recovery and data extraction, essential for accurate communication in OFDM systems. While FFT and IFFT are effective, they pose computational challenges, particularly in high-speed communication systems. To address this limitation, machine learning and deep learning techniques are being explored to replace the conventional FFT process with more efficient alternatives. By leveraging machine learning, it is possible to develop models that can perform the FFT operation with similar or improved performance compared to conventional methods. Specifically, feed forward neural networks can be trained to learn the complex transformations involved in the FFT process. These neural networks are capable of processing large amounts of data and can adapt to various OFDM system configurations. The use of machine learning and deep learning in OFDM receivers offers the potential for significant reductions in computational complexity while maintaining or even enhancing the accuracy of the system.

2 Deep Learning approach for FFT

While various machine learning algorithms could be employed, deep learning, particularly using feed forward neural networks, has been chosen for several reasons:

1. Complexity of Patterns:
   Feed forward neural networks excel in capturing intricate patterns and relationships in the data, which might be challenging for traditional machine learning models to learn.

2. Automatic Feature Extraction:
   Deep learning models automatically learn features from the data, eliminating the need for manual

feature engineering. This is particularly advantageous when dealing with complex data such as signals in OFDM systems.

3. Scalability
Deep learning models scale effectively with large datasets. They can handle vast amounts of data efficiently, making them well-suited for applications where large datasets are common.

4. Flexibility:
Neural networks are highly flexible and can be adapted to different problem domains and data types. This adaptability makes them suitable for a wide range of applications, including OFDM receivers.

5. Improved Performance:
Deep learning models often outperform traditional machine learning models, especially when dealing with large datasets and complex tasks such as signal processing. The ability of deep learning models to learn complex transformations and relationships in the data leads to improved performance in various applications, including the Fast Fourier Transform operation in OFDM receivers.

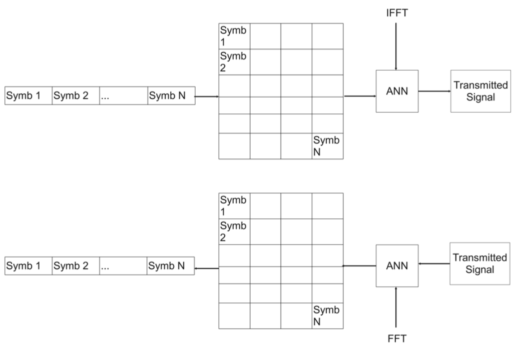

Figure 1: Deep Learning Approach in OFDM

3 Methodology

In OFDM systems, when a time-domain signal undergoes transformation into the frequency domain using the Discrete Fourier Transform (DFT) process, it produces two parts: the real and imaginary values of the frequency domain signal. Traditionally, this transformation is achieved using the Fast Fourier Transform (FFT) algorithm. In this project, we explore an alternative approach where a neural network is trained to predict these real and imaginary parts in the frequency domain directly from the time-domain signal. This task is framed as a regression problem, where the neural network learns to map time-domain signals to their corresponding real and imaginary parts in the frequency domain. This approach aims to improve the efficiency and accuracy of the FFT process in OFDM receivers.

$$DFT(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-\frac{2\pi i}{N} kn} \tag{1}$$

Our objective is to develop a neural network capable of directly computing the Discrete Fourier Transform (DFT) of a time-domain signal, without relying on traditional mathematical equations like the one above. More specifically, we aim to create a neural network for the regression task of calculating the real and imaginary components of a time-domain signal.

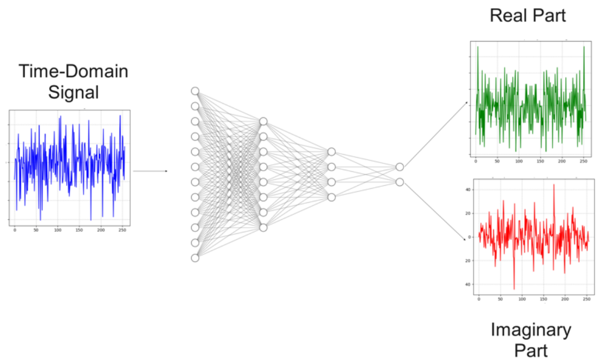

Figure 2: Input and Outputs of The Network

3.1 Dataset

To train the neural network for computing the Discrete Fourier Transform (DFT) of time-domain signals, a dataset comprising time-domain signals and their corresponding frequency-domain representations is required. The dataset consists of 100,000 samples, each comprising time-domain signals with a dimension of 256 samples. These time-domain signals are generated using a normal distribution with zero mean and unit variance. Subsequently, the Fast Fourier Transform (FFT) is applied to these time-domain signals to obtain their frequency-domain representations. The real and imaginary parts of the resulting frequency-domain representations are then extracted and used as the output data for the neural network.

3.2 Network Architecture

The neural network architecture employed for this task is a feed forward neural network specifically designed to directly compute the Discrete Fourier Transform (DFT) of a time-domain signal. The network consists of an input layer that accepts time-domain signals of dimension followed by two hidden layers. Each hidden layer is a fully connected layer with ReLU activation function. The output layer comprises two branches: one for the real part and another for the imaginary part of the frequency-domain signal. Also, because it is a regression task the last layer do not have any activation function.

3.3 Parameters fine tuning and optimization

To optimize the parameters of the neural network model, a grid search approach was employed. The optimization process involved tuning three main hyperparameters: the number of epochs, the learning rate, and the choice of optimizer. For each combination of hyperparameters, the model was trained and validated using mean squared error (MSE) loss as the criterion. The grid search explored different values for the number of epochs (50, 75, and 100), learning rate (0.01, 0.001, and 0.0001), and optimizer (Adam, RMSprop, and Adagrad). After training, the average validation loss was calculated, and the set of hyperparameters that resulted in the lowest validation loss was selected as the best set of parameters for the model. further parameters such as the number of layers and the number of neurons in each layer can be added to the grid search to explore the parameter space more comprehensively and achieve even better results.

4 Results and Discussion

The results of the training process demonstrate that the simple feed forward neural network architecture is capable of computing the Fast Fourier Transform (FFT) faster than traditional FFT methods while achieving a very low mean squared error (MSE). By utilizing the grid search approach to optimize hyper-parameters such as the number of epochs, learning rate, and choice of optimizer, the neural network was able to efficiently learn the complex transformations involved in the FFT process. The validation loss, measured by MSE, was significantly reduced, indicating that the neural network accurately predicts the real and imaginary parts of the frequency-domain signal. This suggests that the neural network not only provides computational advantages in terms of speed but also maintains high accuracy in comparison to traditional FFT methods.

5 Conclusions

we explored the application of machine learning techniques, specifically a feedforward neural network, for computing the Fast Fourier Transform (FFT) in Orthogonal Frequency Division Multiplexing (OFDM) receivers. By training a simple feedforward neural network to directly compute the FFT of time-domain signals, we demonstrated that the neural network architecture can compute the FFT faster than traditional FFT methods while achieving a very low mean squared

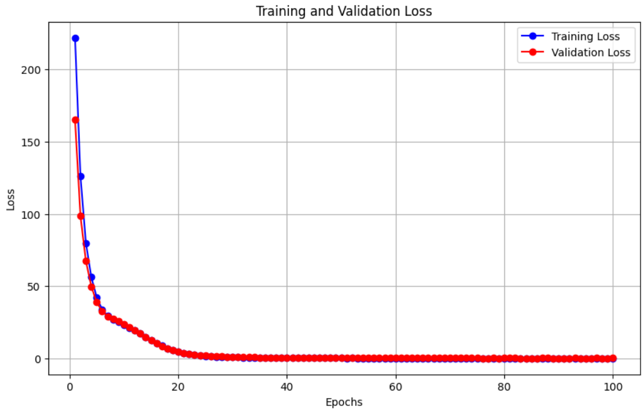

Figure 3: Training and Validation Loss

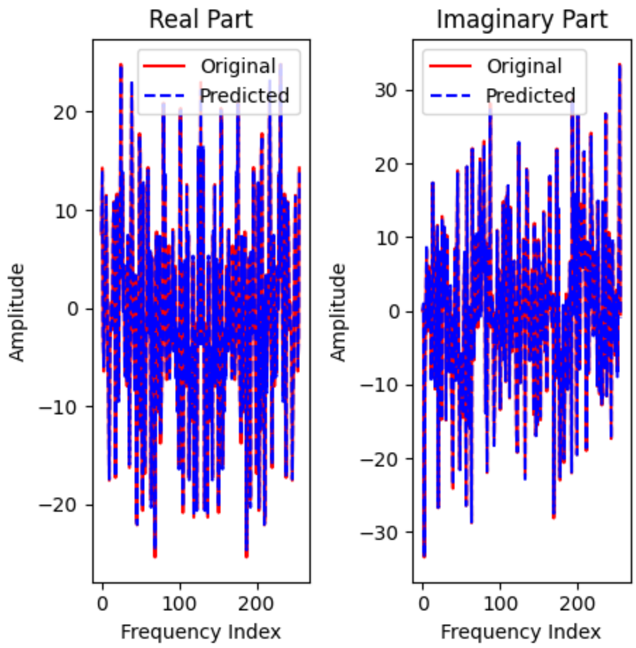

Figure 4: Original Signal and Predicted Ones

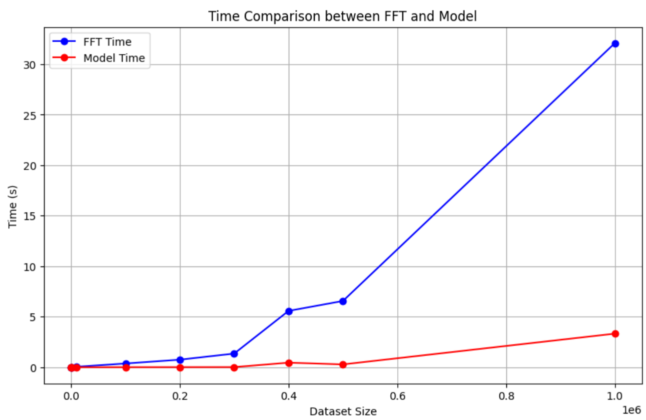

Figure 5: Training and Validation Loss

error (MSE). The grid search approach allowed us to optimize hyperparameters effectively, resulting in improved performance in terms of both speed and accuracy. Overall, this project highlights the potential of machine learning techniques to enhance signal processing tasks in communication systems, offering computational advantages without compromising accuracy. Moving forward, further research could explore more complex neural network architectures and additional optimization techniques to further improve the efficiency and accuracy of signal processing tasks in OFDM systems.

Table 1: Hyperparameters Used for Training

Parameter	**Value**
Epochs	100
Learning Rate	0.001
Optimizer	Adam
Batch Size	1024
Activation functions	RELU
Number of Hidden Layers	2
Neurons per Hidden Layer	512