High Resolution Spectral Estimation
Bachelor Thesis Project

Omkar Nitsure
Guide: Prof. Satish Mulleti

Electrical Engineering, IIT Bombay

Nov 20, 2024

Github Link: https://github.com/omkar-nitsure/HRSE/tree/main

Introduction

- Frequency estimation from samples corrupted by noise is a fundamental challenge
- Algorithms like MUSIC and ESPRIT can be used for resolution
- Resolution capacity is limited by noise level and number of measurements
- Methods that require fewer samples are highly desirable as acquiring samples is expensive

Problem Statement

- Given: Set of samples of a signal acquired through a series of sensors (radar)
- Solve: Spectral composition of the signal
- Assumption: Spectrum non-zero only for 2 frequencies
- Smallest resolution: 1/10th of the theoretical limit

Theoretical limits

- N : Number of samples available through sensors
- Assuming, the signal consists of 2 frequencies, they can be successfully resolved using methods like ESPRIT if the separation follows:

$$|f_1 - f_2| = \frac{1}{N}$$

Experiment Setup

- We start with $N = 50$ samples (model input)
- We use machine learning models to predict $M = 100$ future samples (model output)
- Concatenate the above 2 to get $N + M = 150$ samples
- Use standard frequency estimation algorithms (ESPRIT) to find frequencies given 150 samples

Purpose

- Due to limited budget of 50 samples we are otherwise restricted to a resolution limit of $\frac{1}{50}$
- If the model accuracy is high, we can reduce the resolution limit to $\frac{1}{150}$

Dataset Generation

Considerations

- We want the model to generalize well to a range of frequency separations above the frequency limit
- We want the model to perform well even for low SNRs
- We don't want the model to overfit the training data

Solutions

- Make sure that the dataset has reasonable proportions of different resolutions
- Train different models for different SNR ranges
- Generate a large enough dataset with sufficient randomness in different signal parameters

Signal formulae

$$x(n) = \sum_{l=1}^{L} a_l e^{j2\pi f_l n} + w(n), \quad n = 1, \ldots, N$$

here, $L = 2, a_1 = a_2 = 1, w(n) \sim \text{Normal}(0, \sigma^2)$

$$\text{SNR} = 10 \log_{10}\left(\frac{|x(n)|_2^2}{N\sigma^2}\right)$$

Frequency Selection

We select frequencies from 4 sets to ensure enough diversity.
Here $\Delta f = \frac{1}{N}$,

- Set 1: 20000 examples such that
  $f_1 \sim \text{Uniform}(0, 0.5 - \Delta f), f_2 = f_1 + 0.5\Delta f$
- Set 2: 20000 examples such that
  $f_1 \sim \text{Uniform}(0, 0.5 - \Delta f), f_2 = f_1 + \Delta f + \epsilon$
  $\epsilon \sim \text{Uniform}(-(f_1 + \Delta f), 0.5 - f_1 - \Delta f)$
- Set 3: 5625 examples where f_1 and f_2 are selected from a grid in the range [0, 0.5]. Grid separation is Δf
- Set 4: 20000 examples such that
  $f_1 \sim \text{Uniform}(0, 0.5 - \Delta f), f_2 = f_1 + k\Delta f$
  $k \in \left[\lceil (-\frac{f_1}{\Delta_f}) \rceil, \cdots, \lfloor \frac{0.5 - f_1}{\Delta_f} \rfloor \right]$
- Set 5: 20000 examples such that, $f_1, f_2 \sim \text{Uniform}(0, 0.5 - \Delta f)$
- Set 6: 20000 examples such that, $f_1 \sim \text{Normal}(0.25, 0.25)$,
  $f_2 \sim \text{Uniform}(0, 0.5 - \Delta f)$

Model Details

We used 2 model architectures,

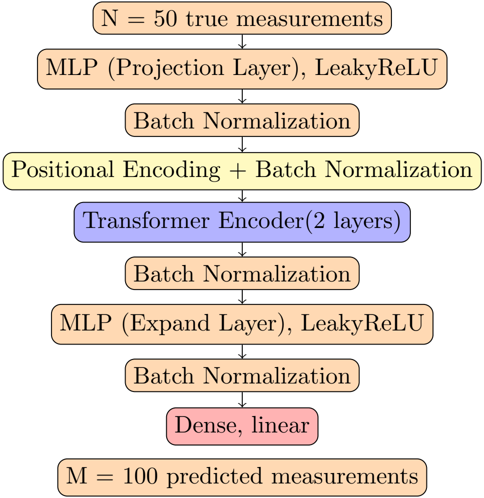
- hybrid bidirectional LSTM-CNN
- Transformer-Encoder - 0.46 million learnable parameters

Loss function and Evaluation metric

$$\text{Loss (L): } \sum_{i=1}^{I} \|\mathrm{x}_{m,i} - G_{\boldsymbol{\theta}}(\mathrm{x}_{a,i})\|_2^2$$

$$\text{Metric: NMSE} = \frac{\frac{1}{K}\sum_{k=1}^{K}(f_k - \tilde{f}_k)^2}{\frac{1}{K}\sum_{k=1}^{K}(f_k)^2}$$

Model architecture for Transformer Encoder



DeepFreq: Problem Formulation

The problem formulation is the same as above, but for the sake of completeness it is given below in the terminology they used

- **Signal Model:** Multisinusoidal signal representation

$$S(t) = \sum_{j=1}^{m} a_j e^{i2\pi f_j t},$$

  where $a_j \in \mathbb{C}$ represents amplitude and phase, $f_j \in [0, 1]$ denotes unknown frequencies, and t is time.

- **Measurement Model:** Observations are given by

$$y_k = S(k) + z_k, \quad 1 \leq k \leq N,$$

  where z_k represents additive noise. The goal is to estimate $f_1, \dots, f_m$ from noisy samples y_k.

Methodology

- **Frequency Representation:** The neural network is trained to approximate a ground-truth frequency representation

$$FR(u) = \sum_{j=1}^{m} K(u - f_j),$$

where K is a narrow Gaussian kernel centered at each frequency f_j.

- **Counting Module:** A convolutional neural network counts the frequency components by analyzing local maxima in the learned frequency representation.
- **Objective:** Minimize the loss function

$$\text{Loss} = \|\text{DeepFreq}(y) - FR(u)\|_2^2\,,$$

where $FR(u)$ is the true frequency representation and $\text{DeepFreq}(y)$ is the network's output.

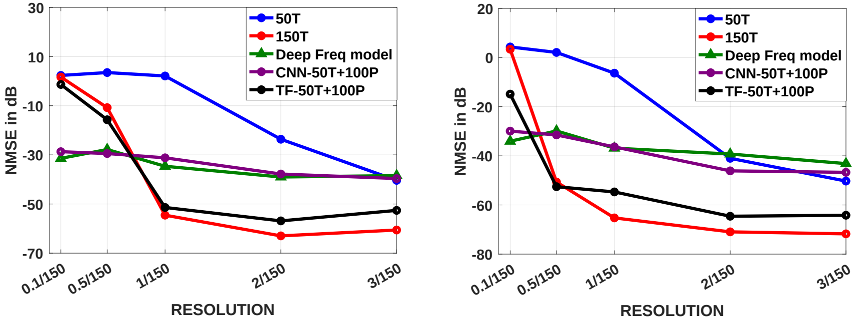
Experimental Setup and Results

- **Chamfer Distance:** To evaluate performance, the Chamfer distance $d(f, \hat{f})$ is calculated between true frequencies $f = (f_1, \dots, f_m)$ and estimates $\hat{f} = (\hat{f}_1, \dots, \hat{f}_{\hat{m}})$,

$$d(f, \hat{f}) = \sum_{f_i \in f} \min_{\hat{f}_j \in \hat{f}} |f_i - \hat{f}_j| + \sum_{\hat{f}_j \in \hat{f}} \min_{f_i \in f} |\hat{f}_j - f_i|.$$

- **Comparison:** DeepFreq performs similarly to the hybrid bidirectional LSTM-CNN model. The Transformer-encoder model performs better than both in high-resolution cases

Results



(a) SNR of 5dB (b) SNR of 15dB

Figure 1: NMSE Vs Reolution for different SNR values

Performance for different SNRs

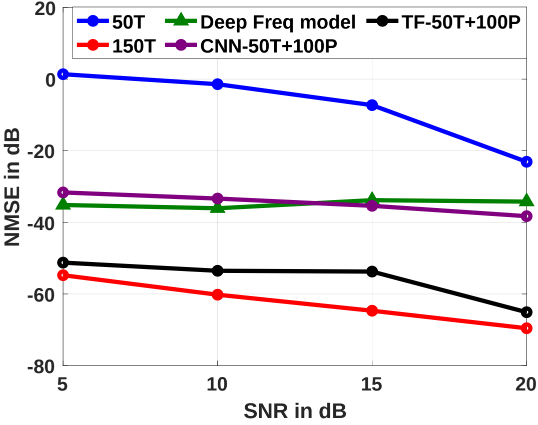


Figure 2: NMSE for resolution of 1/50

Signal Model for Finite Rate of Innovation

- **Signal Model:** The FRI signal is represented by a periodic stream of Dirac pulses:

$$x(t) = \sum_{k=0}^{K-1} a_k \delta(t - t_k),$$

where a_k are the amplitudes and t_k the locations of the Dirac pulses.

- **Acquisition Model:** The continuous signal is sampled with kernel $\phi(t)$:

$$y[n] = \langle x(t), \phi(t/T - n) \rangle = \sum_{k=0}^{K-1} a_k \phi\left(\frac{t_k}{T} - n\right).$$

Conversion to Sum of Exponential

- Let,

$$s[m] = \sum_{n=0}^{N-1} c_{m,n} y[n]$$

- Exponential Reproducing Kernel and Frequency Separation:

$$\sum_{n \in \mathbb{Z}} c_{m,n} \varphi(t-n) = \exp(j\omega_m t) \quad \text{where} \quad \omega_m = \omega_0 + m\lambda$$

- Then we can write in terms of the Sum of Exponentials:

$$s[m] = \sum_{k=0}^{K-1} b_k (\mu_k)^m$$

- Perfect Prediction in Noise-Free Case: For any $s[m]$, there exists a set of coefficients $\{c_k\}_{k=1}^{K}$ such that $s[m] = \sum_{k=1}^{K} c_k s(m-k)$.

Learning-Based FRI Reconstruction

- **Deep Unfolded Projected Wirtinger Gradient Descent (PWGD):** Learns parameters of the denoising process to solve for $\{t_k\}$ under noisy conditions.
- **FRIED-Net:** Encoder-decoder model for FRI reconstruction, useful when kernel $\phi(t)$ is unknown. Consists of:
  - *Encoder:* Estimates Dirac locations directly from noisy samples.
  - *Decoder:* Resynthesizes samples $y[n]$ based on estimated parameters:

$$y[n] = \sum_{k=0}^{K-1} a_k \phi\left(\frac{t_k}{T} - n\right).$$

Loss Functions

- **Unfolded PWGD Loss:** Minimizes the error between denoised matrix $\hat{S}$ and the true annihilating filter h:

$$L(S) = \|\hat{S}h\|_2^2 + \alpha e^{-\beta\|\hat{S}\|_F^2}.$$

- **FRIED-Net Loss:** Combines errors in reconstructed locations $\{t_k\}$ and samples $\{y[n]\}$:

$$L(y,t) = \sum_n (y[n] - \hat{y}[n])^2 + \gamma \sum_k (t_k - \hat{t}_k)^2.$$

Method and Dataset

- We use $N = 21, M = 39$
- We use the analytical formula given above for $s[m]$ to compute the future noiseless M samples which are then used for training the model
- All samples are scaled down using the maximum value of the analytical samples achieving better training convergence

Results

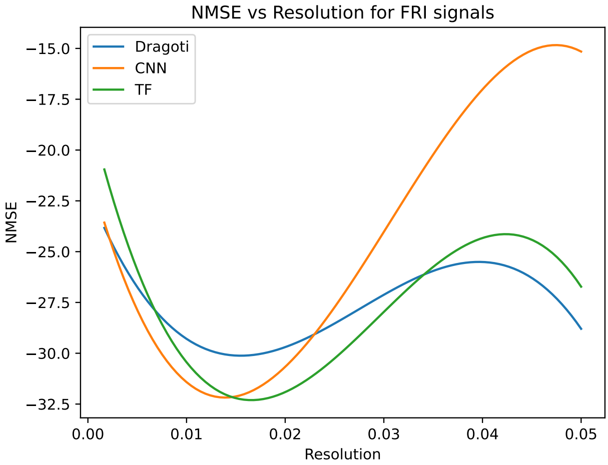


Figure 3: NMSE for resolution of 1/50

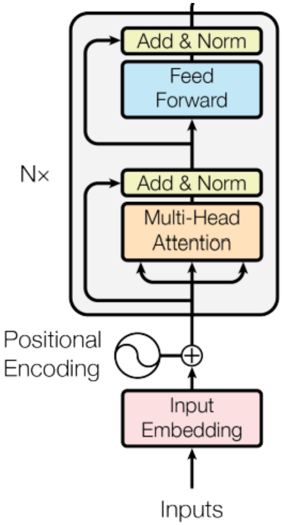
References

- G. Izacard, S. Mohan, and C. Fernandez-Granda, "Data-driven Estimation of Sinusoid Frequencies," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.
- V. C. H. Leung, J.-J. Huang, Y. C. Eldar, and P. L. Dragotti, "Learning-based Reconstruction of FRI Signals," *IEEE Transactions on Signal Processing*, vol. 71, pp. 2564–2578, 2023.
- S. K. Dondapati, O. Nitsure, and S. Mulleti, "Super-Resolution via Learned Predictor," arXiv preprint arXiv:2409.13326, 2024.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 5998–6008, 2017.
- J. Alammar, "The Illustrated Transformer," *Jay Alammar's Blog*, [Online]. Available: `https://jalammar.github.io/illustrated-transformer/`.

Questions/Suggestions

Thank you!

Transformer Encoder: Detailed Breakdown



- Feedforward is an MLP layer (The middle layer has 1024 Neurons)
- We have used a learnable matrix as the positional encoding
- **Add & Norm** is the standard residual connection followed by Layer Normalization
- Multi-Head Attention: It has multiple attention heads (we used 8)

How does the Self-Attention Work?

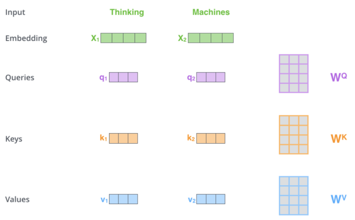


Figure 4: qkv computations

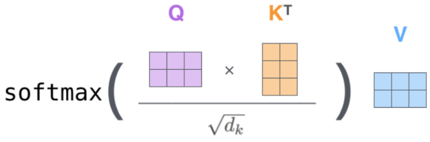


Figure 5: self-attention formula

- W^Q, W^k, and W^v are learnable projection matrices
- Dot product between Q and K measures how relevant different K are for the Q
- Scaling of $\sqrt{d_k}$ is used to prevent SoftMax values from saturating
- SoftMax gives the probability distribution and the corresponding V are added in that proportion