Title: Comparative Study of TUNER and SIREN Architectures

Presented by: Yanet Tesfaye

Under mentorship of *Professors Tiago Novello & Lúiz Velho*

Date: *July 2025*

GOAL : Compare TUNER and SIREN in:

Architecture

Initialization

Training approach

Remaining Open Questions

Sinusoidal neural networks have gained traction as powerful implicit neural representations (INRs) to model low-dimensional signals such as images, audio, and 3D shapes.

IMAGES

AUDIO

3D SHAPES

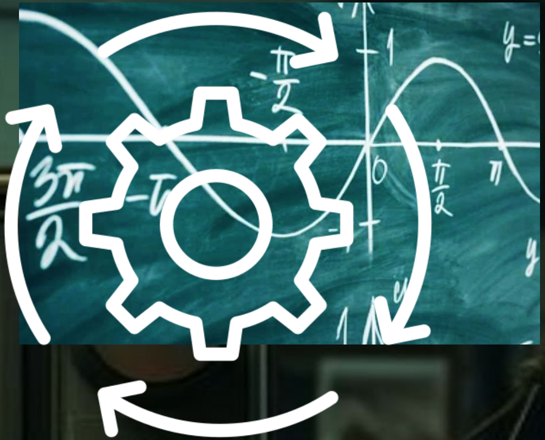
MORE…

Traditional neural networks suffer from a spectral bias. Thus, Sinusoidal activation functions overcome this limitation by inherently modeling signals with high-frequency content.

- **Challenges**: initializing and training sinusoidal networks is difficult and often empirical.

The initialization directly influences the network's ability to represent the frequency spectrum of the signal, and poor initialization often leads to noisy or overfitted outputs. Both **SIREN** and **TUNER** attempt to address these challenges but with different approaches and theoretical underpinnings.

Overview of SIREN Architecture



ACTIVATION FUNCTION

Standard multilayer perceptron (MLP) with sine activation functions.

WEIGHT INITIALIZATION

Weight initialization heuristic scales weights by ω_0 (typically 30) to stabilize gradients.

MAPPING

Input layer maps inputs through sine functions with randomly chosen frequencies in a range.

OBSERVED PROBLEM

Empirical initialization can lead to overfitting or noisy high-frequency artifacts.

- The key innovation in SIREN is the initialization scheme that scales the weights of the first layer to avoid vanishing or exploding gradients.
- Specifically, weights of the first layer are initialized uniformly in the range $[-1/\omega_0, 1/\omega_0]$.
- Despite these innovations, SIREN's initialization remains empirical: the choice of ω_0 and weight ranges is somewhat heuristic and sensitive to the specific dataset and signal
- Additionally, SIREN lacks explicit frequency control during training, causing the network to sometimes generate frequencies outside the Nyquist limit, leading to artifacts.

Overview of TUNER-Theoretical Foundation and Approach

TUNER builds on sinusoidal MLPs by applying Fourier series theory.

Theorem 1: Each hidden neuron expands into a sum of sinusoids with integer combinations of input frequencies.

- Input frequencies are fixed integers, aligning with Fourier basis, and frozen during training.

- **TUNER** (TUNing sinusoidal nEtwoRks) builds on sinusoidal MLPs by introducing a rigorous theory to explain how frequencies emerge through layer compositions.

  - Unlike **SIREN's** random **uniform initialization**, **TUNER** fixes the input frequencies (freezing them during training) to guarantee that the network covers the desired frequency spectrum in a controlled way.
  - Moreover, TUNER introduces a **band limit control mechanism** based on bounding the hidden layer weights during training.

  - By clamping these weights, **TUNER** prevents the generation of spurious high frequencies beyond the **Nyquist limit**, thus reducing noise and overfitting.

Detailed Architecture Differences

SIREN

1. Standard MLP with sinusoidal activations applied at every layer.
2. Input layer uses sine activations with frequencies randomly sampled from a uniform range.
3. Weight initialization is empirical and static — no frequency control during training.
4. No explicit frequency bounding — network frequencies can drift during optimization.

TUNER

1. Input frequencies are initialized as fixed integers aligned to the Fourier basis and frozen during training.
2. Hidden weights are clamped to control amplitude of frequencies, enforcing band limit constraints
3. Training uses a new regularization term that keeps hidden weights small, preventing frequency explosion.
4. The architecture uses trigonometric identities and Fourier analysis to inform its design and training.

Initialization Comparison

➔ Initialization is the cornerstone of effective sinusoidal networks.

SIREN Initialization

1 Chooses input frequencies uniformly at random from a fixed range of :

IMPORTANT

Ignore some important frequencies

$$[-B, B]$$

Wide range of frequencies

Doesn't guarantee full coverage of frequency

Undesirable high frequencies

Initialization Comparison

➔ Initialization is the cornerstone of effective sinusoidal networks.

TUNER Initialization

2 TUNER Initialization instead samples input frequencies as integer multiples in a limited band of:

Equivalent to sampling from a discrete Fourier basis

$$[-b, b]$$

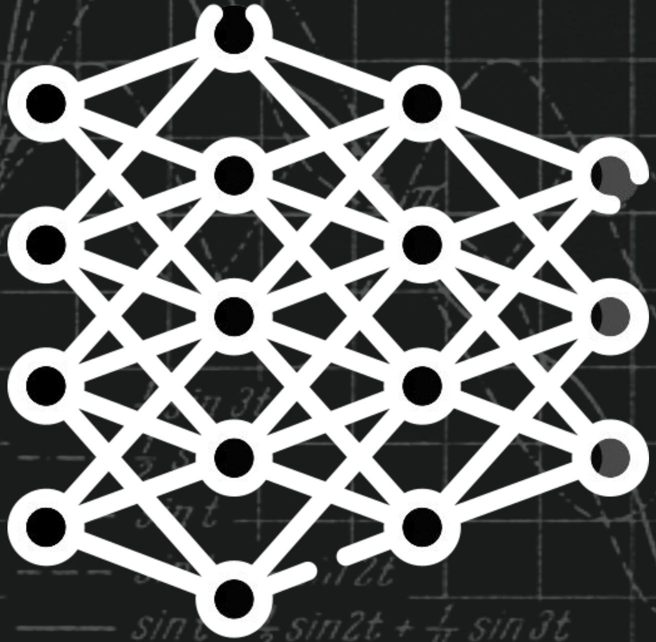
Ensures complete spectral coverage

Splits the input frequencies into “low” and “high” frequency groups

Input frequencies are kept in the low band to reduce overfitting.

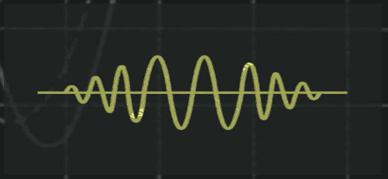
Training Differences and Bandlimit Control

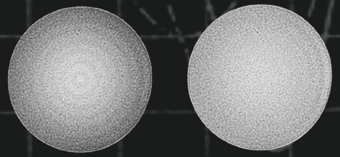
Training in SIREN



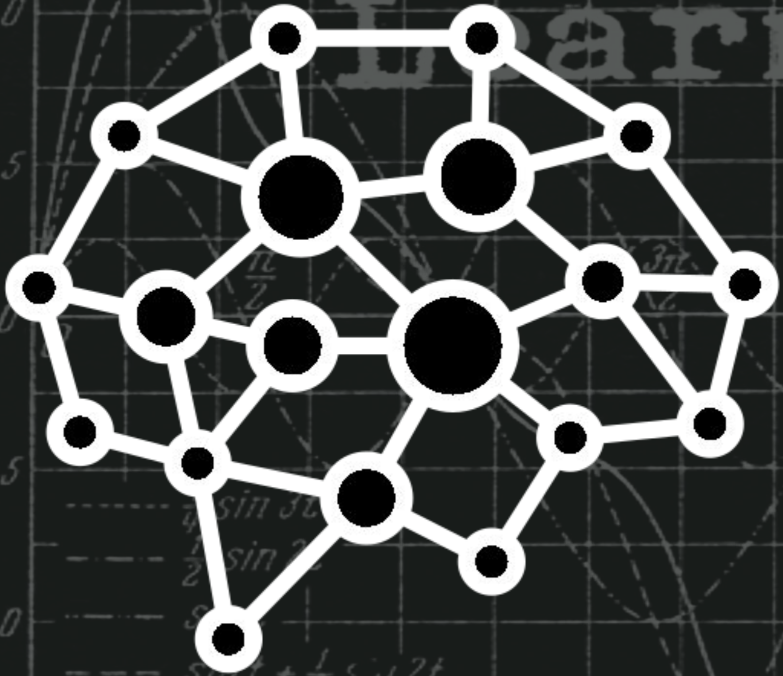
1. Hidden weights can grow arbitrarily large, which increases the amplitude of high-frequency components, often beyond the Nyquist limit.

2. This uncontrolled growth causes ringing artifacts and noisy reconstructions, requiring early stopping





Training in TUNER



- Employs a weight clamping mechanism where weights in the hidden layers are limited to a predefined range
- This limits the amplitudes of higher frequency components generated by the network
- Introduces a regularization term based on the upper bounds of amplitudes (derived from theory)

REGULARIZATION TERM :

$$L_{\mathrm{reg}} = \sum |c_j|$$

Penalize large weight magnitudes

Training thus balances fitting accuracy with frequency control, reducing overfitting and noisy artifacts.

Potential Problems Both Methods Leave Unsolved

Shallow Architecture Only

Experiments focused on 3-layer MLPs.

Scalability to High Dimensions

Need for Task-Specific Tuning

Frequency Coverage Trade-off (TUNER)

Summary and Final Remarks

- SIREN introduced sinusoidal activations with heuristic initialization, enabling high-frequency modeling but with instability.
- TUNER adds a theoretical framework grounded in Fourier series, with principled initialization and frequency bounding.
- TUNER achieves more stable training, faster convergence, and cleaner reconstructions
- However, both models leave room for further improvement in flexibility, scalability, and understanding of frequency learning dynamics.

THANK YOU FOR LISTENING AND FOR YOUR GUIDANCE !