Proceedings Track

Periodic Signal Recovery with Regularized Sine Neural Networks

Anonymous author 1 ANONYMOUS@ANONYMOUS.COM
Anonymous author 2 ANONYMOUS@ANONYMOUS.COM
Anonymous author 3 ANONYMOUS@ANONYMOUS.COM
Common affiliation of authors

Editors: List of editors' names

Abstract

We consider the problem of learning a periodic one-dimensional signal with neural networks, and designing models that are able to extrapolate the signal well beyond the training window. First, we show that multi-layer perceptrons with ReLU activations are provably unable to perform this task, and lead to poor performance in practice even close to the training window. Then, we propose a novel architecture using sine activation functions along with a well-chosen non-convex regularization, that is able to extrapolate the signal with low error well beyond the training window. Our architecture is several orders of magnitude better than its competitors for distant extrapolation (beyond 100 periods of the signal), while being able to accurately recover the frequency spectrum of the signal in a multi-tone setting.
Keywords: Machine Learning, Periodic Signals, Frequency Estimation.

1. Introduction

Real-world signals, such as weather forecasts, biological data or financial indicators, often exhibit periodic patterns. These patterns are notoriously difficult to capture with standard neural networks such as ReLU networks (Parascandolo et al., 2017; Eger et al., 2019), thus making deep learning approaches rather unsuccessful for these prediction tasks. While most of the literature focused on using activation functions that exhibit a periodic behavior (Parascandolo et al., 2017; Eger et al., 2019; Ziyin et al., 2020; Mehta et al., 2021), our analysis shows that this approach is, in essence, *necessary yet insufficient*. More precisely, we show that ReLU networks are fundamentally unable to learn periodic signals, and changing the activations to sine functions requires additional work to ensure proper training. In this work, we propose a novel regularization for sine neural networks that is able to accurately learn the frequency spectrum of a target signal, and thus extrapolate it well beyond the training window.

Problem setup. We consider the problem of recovering a periodic function given observations only on a bounded interval. Under the condition that at least one period is observed, this problem is always well-defined, however the recovery of the periodic function on the full real line requires one to uncover the underlying symmetry, which can be significantly harder than fitting the data on the observed interval.

For all the following, let $T \in \mathbb{R}_+^*$ be a fixed but unknown period, and $R \in \mathbb{R}_+^*$ be a fixed window size. For all $t \in \mathbb{R}_+^*$, let $\mathcal{F}_t \subseteq (\mathbb{R} \to \mathbb{R})$ be the set of t-periodic functions, i.e. $f \in \mathcal{F}_t \Leftrightarrow \forall x \in \mathbb{R},\ f(x+t) = f(x)$. Note that these sets of functions are not disjoint. We

© 2022 A.a. 1, A.a. 2 & A.a. 3.

will assume in all the following that at least one period has been observed, that is to say $R \geq T$. Without this condition, recovering the entire function is clearly impossible, and this condition alone is sufficient to make the problem well defined: If $f^* \in \mathcal{F}_T$ with $T \leq R$, then (up to zero-measure terms) the squared-error minimizer is unique

$$\operatorname*{argmin}_{f \in \cup_{t \leq R} \mathcal{F}_t} \mathbb{E}_{x \sim \mathcal{U}(0,R)} \left[(f(x) - f^*(x))^2 \right] = \{f^*\}$$

The difficulty of this problem does not come from a plethora of indistinguishable minima, as is customary in machine learning, but from the unusual structure of the assumption. Although each $\mathcal{F}_t$ is a vector space, their union is not, for the sum of two periodic functions is not periodic in general. It also lacks an easily exploitable convexity to treat it as an optimization problem that we could easily solve. Moreover, ignoring the periodicity hypothesis and just fitting a function $g : [0, R] \to \mathbb{R}$ to the segment, with a neural network for instance, renders the problem ill-defined, for the loss is supported only on the segment and does not specify how to extend the function to the entire real line.

2. Proposed architecture and training procedure

2.1. Regular Sine-based neural networks

We experiment with two-layer networks with sine as activation ($\sin : \mathbb{R} \to [-1, +1]$). For a width $m \in \mathbb{N}^*$, we initialize independently at random $\omega_i \sim \mathcal{N}(0, 1)$, $a_i \sim \mathcal{N}(0, 1/\sqrt{m})$ and $b_i \sim \mathcal{N}(0, 1/\sqrt{m})$, to get weights $(\omega, a, b) \in \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m$. This ensures $\sum_i \mathbb{E}[a_i^2] = 1 = \sum_i \mathbb{E}[b_i^2]$, so the initial response is bounded. We train (ω, a, b) by gradient descent to minimize the empirical loss, using as prediction function $F_{\omega,a,b} : x \mapsto \sum_{i \in [m]} a_i \sin(\omega_i x) + b_i \cos(\omega_i x)$.

2.2. Modified initialization and regularization

This minimal modification of the traditional multi-layer perceptron has had little success, because it does not seem to converge to approximations as good as the ReLU-based neural networks. We show that by carefully altering the initialization procedure and regularization, this architecture is capable of distant extrapolation of periodic functions, beating the ReLU-based networks by several orders of magnitude.

To ensure diversity of the initial frequency weights, we initialize them at regular intervals $\omega_i = i\pi/R$ for $i \in [m]$ (there is no need for negative frequencies by symmetry). We initialize the amplitude weights at zero $(a, b) = (0, 0)$. To promote sparsity in the amplitudes, and avoid interferences between very close but distinct frequencies, we use a non-convex sparsity-promoting penalty (with $\varepsilon =$ 1e-20 a small constant for differentiability)

$$\mathcal{R} : (a, b) \mapsto - \sum_{i \in [m]} \exp\left(- \sqrt{a_i^2 + b_i^2 + \varepsilon} \right)$$

Note that we are not interested in finding the minimum to a different (regularized) loss function, we are merely trying to steer the trajectory of weights toward relatively sparser amplitudes *during* training. For this reason, we instead use the time-dependent loss $(t, w, a, b) \mapsto \mathcal{L}_0(w, a, b) + e^{-t/\tau} \mathcal{R}(a, b)$ (with $\tau = 10$ in experiments), that is a weighted

Proceedings Track

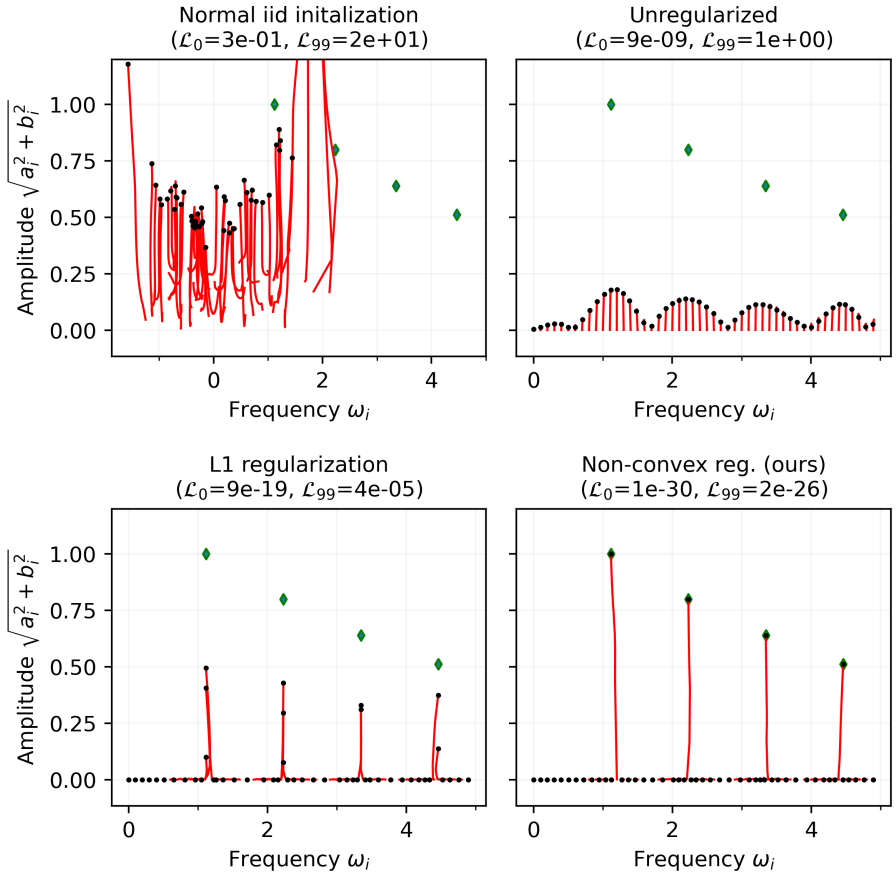

Figure 1: Neuron trajectories in the Fourier half-plane. Black dots indicate the final position of each neuron, red trails depict the trajectory of each neuron during training. Green diamonds indicate an optimal configuration representing the signal exactly (though this parametric global minimum is not unique)

sum of the empirical loss $\mathcal{L}_0$ and our regularizer $\mathcal{R}$, but whose weight tends to zero during training, such that we truly converge to a minimum of the empirical loss alone when possible.

In the following sections, we explain the reasons that motivated these choices, starting with the need for a different periodicity-compatible activation function, and then how to fix the training problems that this change induces.

2.3. Limits of universal approximation on compacts, the need for sines

A historically strong argument in favor of the representation of functions as neural networks is the theorem of universal approximation on compacts (Cybenko, 1989; Barron, 1993; Leshno et al., 1993), stating that for any continuous function with a compact domain, there exists functions arbitrarily close to it that are representable as a neural network, provided only

that the activation function is non-polynomial. This property is not sufficient for periodic recovery, as the followng proposition shows.

Proposition 1 *If a function $f : \mathbb{R} \to \mathbb{R}$ is representable as a multi-layer perceptron with ReLU activations and finitely many weights, and if f is periodic, then f is constant.*

Proof Such a function is piecewise linear with finitely many pieces (see e.g. Hanin and Rolnick, 2019). Thus there exists a constant $B > 0$ such that it is affine on $[B, +\infty[$. If it is both periodic and affine on that interval, then it must be constant on that interval, thus it is constant on $\mathbb{R}$ by periodicity. ■

This proposition does not constitute a problem in itself, because it remains possible that the learned function is not perfectly periodic, yet remains a very good approximation for a large number of periods after the observed interval, which would suffice in all practical applications. However, our experiments demonstrate that such networks fail to extrapolate to even a single period outside the training interval.

2.4. Problems of the regular sine-based architecture

As presented in the previous section, the sine-based architecture is strikingly similar to a two-layer perceptron with only a modification in activation functions, and this is sufficient to fix the representation problem, all periodic functions are approximable on the whole real line by networks of this form. However, experiments show that training this modified architecture as-is still fails to extrapolate outside the training interval. We identify several problems and propose modifications to the training procedure to fix each of these issues.

Diagnostic of ill-convergence. Figure 1 depicts the trajectories of neuron weights over time with two different initializations. A weight $(w, a, b) \in \mathbb{R}^{m\times 3}$ is depicted as a set of m points $(w_i, a_i^2 + b_i^2) \in \mathbb{R} \times \mathbb{R}_+$, together with the corresponding red trail for their evolution over time. The green diamonds correspond to an optimal set of weights, exactly representing the signal. As seen in the first case of Figure 1, independent initialization of frequency weights ($w \in \mathbb{R}^m$) with a normal distribution barely ever produces very high frequencies, which become harder for the network to learn (a long "distance" has to be traveled by the weights), a more diverse initialization appears preferable.

Additional regularization to avoid interferences. The trajectories of gradient descent have several times been linked with a form of "ℓ_2-inertia", see for instance Gunasekar et al. (2018). Indeed in Figure 1, we see that many neurons have a small but non-zero amplitude. While this may not be a problem in the training interval, it tends to cause unwanted wiggling outside the training interval, because these frequencies are usually not rationally linked and thus do not produce exactly periodic functions but only functions that look approximately periodic on the training interval, see for instance learned signals in Figure 2. This issue can be fixed by promoting sparsity in the amplitudes, but some care must be taken to not introduce a phase bias by the choice of regularization. Indeed, a change of phase in the signal corresponds to a rotation in the (a, b) plane. Therefore, to avoid the phase-bias, the regularization chosen must depend on $r_i^2 = a_i^2 + b_i^2$ but not a_i and b_i individually. As shown in Fig. 1, an ℓ_1 regularization $\sum_i r_i$ does induce sparsity, but does not distinguish between

Proceedings Track

two very similar neurons, which causes issues on large extrapolation intervals, which are of particular interest in these experiments (see loss values in Fig. 1). To avoid these interferences, we use the non-convex penalty $\mathcal{R}(a, b) = -\sum_i \exp(-r_i)$, which even for $\omega_i = \omega_j$, forces either r_i or r_j to be null, where the ℓ_1 penalty allows any combination with constant sum.

3. Experiments

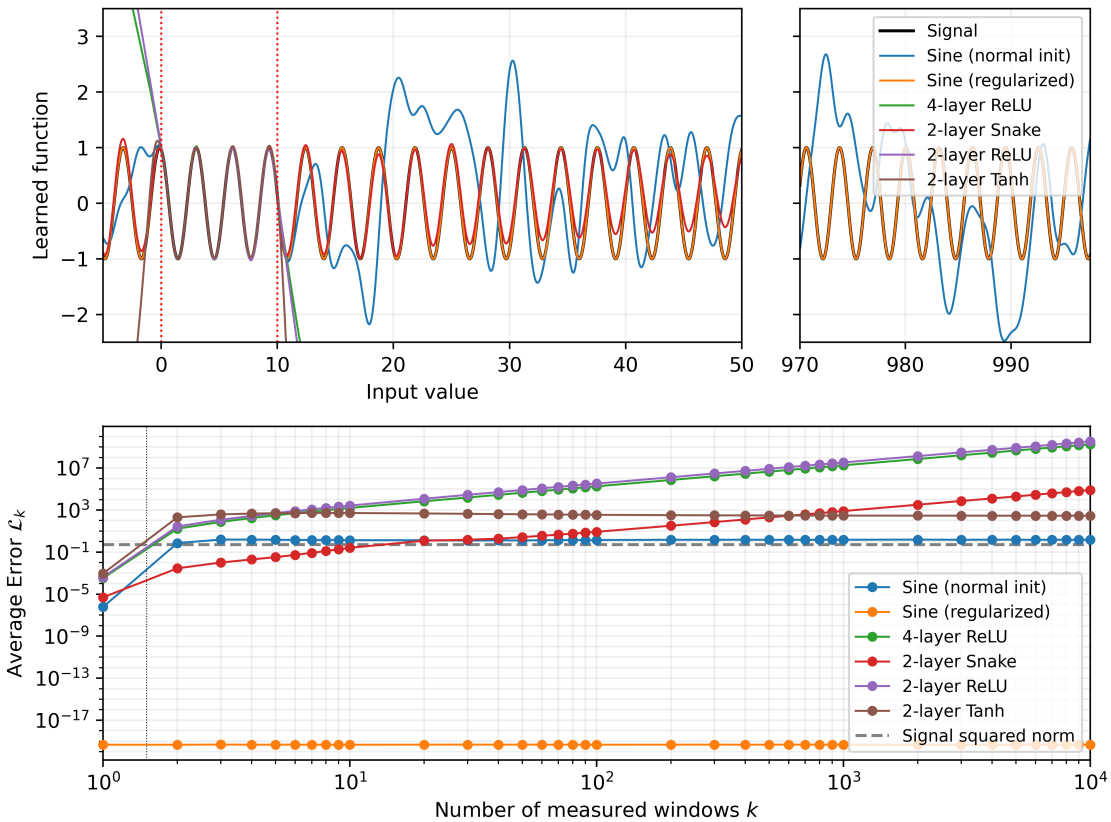

Figure 2: Learned function and corresponding error for each architecture (single tone signal).

Evaluation methods. We use $R = 10$ as window size for our experiments. For each experiment, we select a period $T \in [0, R]$, a T-periodic function $f^* \in \mathcal{F}_T$, a number of observed samples $n \in \mathbb{N}^*$. We then sample independently uniformly at random on $[0, R]$ a collection of n samples $(x_i)_{i\in[n]} \in [0, R]^n$. Along with these inputs, algorithms are provided with the noiseless response $(y_i = f^*(x_i))_{i\in[n]} \in \mathbb{R}^n$. After training, we evaluate the empirical error $\mathcal{L}_0 : f \mapsto \frac{1}{n}\sum_i (f(x_i) - y_i)^2$, and the k-windows expected error $\mathcal{L}_k : f \mapsto \mathbb{E}_x[(f(x) - f^*(x))^2]$ where $x \sim \mathcal{U}(0, k\,R)$, for various values of $k \in \mathbb{N}^*$. In similar settings in machine learning, the empirical error $\mathcal{L}_0$ is referred to as "train loss", and the 1-window expected error $\mathcal{L}_1$ as the

"test loss", for its underlying distribution corresponds to that of the samples. However, the performance we are truly interested in is the k-windows expected error $\mathcal{L}_k$ for large values of k (typically $k \geq 10$ at the very least for simple signals), for which we can really claim that the periodicity assumption has been exploited.

Baseline architectures evaluated We experiment with multi-layer perceptrons with ReLU activations ($(\cdot)_+ : x \in \mathbb{R} \mapsto \max(0, x)$). For a number of hidden layers $L \in \mathbb{N}^*$, and a width $m \in \mathbb{N}^*$, we initialize independently at random $\theta_i^0 \sim \mathcal{N}(0,1)$ for $i \in [m]$ for the first layer, $\theta_{i,j}^k \sim \mathcal{N}(0,1)$ for $(i,j) \in [m] \times [m]$ for layer $k \in \{1, \cdots, L-1\}$, and $\theta_i^L \sim \mathcal{N}(0, 1/\sqrt{m})$ for $i \in [m]$ for the last layer. For an input $x \in \mathbb{R}$, we set the initial activation $z^0 = x$, then recursively $z_i^{k+1} = \max(0, \theta_i^k \cdot z^k)$ for $k < L$, and use as prediction function $F_\theta(x) = \sum_i \theta_i^L z_i^L \in \mathbb{R}$. We then train the weights $\theta \in \mathbb{R}^m \times (\mathbb{R}^{m \times m})^{L-1} \times \mathbb{R}^m$ by gradient descent (Adam optimizer, step size 1e-4) to minimize the empirical loss. We also perform experiments with other non-linearities such as tanh, and the snake $x \mapsto x + \frac{1}{a}\sin^2(ax)$ activation (Ziyin et al., 2020) with $a = 27.5$ (authors recommend $a \in [5, 50]$).

Single-tone periodic recovery We set $\omega^* = 2$ and signal $f^* : x \mapsto \cos(\omega^* x + \pi/12)$. We use $n = 1000$ samples. Figure 2 shows the performance of each method, along with the function learned. All methods achieve reasonably low error on the training set, and comparable "test" error $\mathcal{L}_1$ (which is almost identical because a thousand-point average on an interval of length 10 is a very good estimate). However, the ReLU architecture fails to uncover the symmetry and produces linear extrapolations outside the training interval.

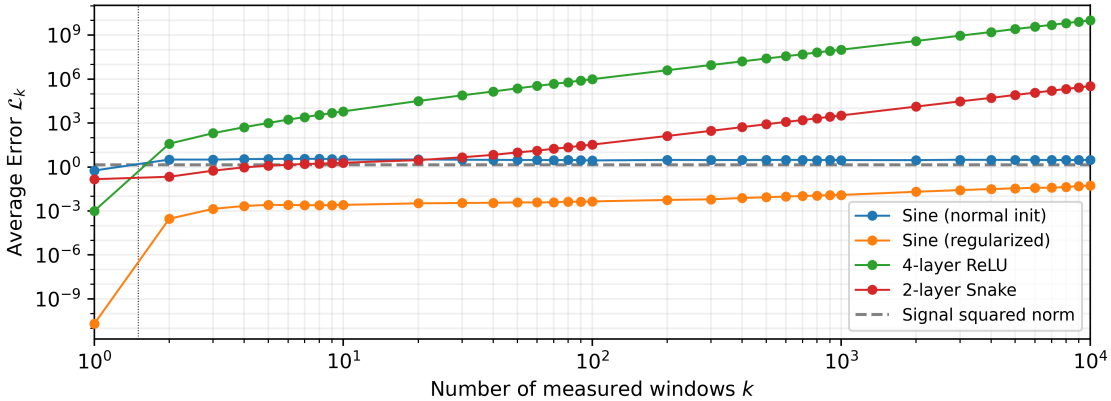

Figure 3: Error of the various architectures with a more complicated signal.

Multi-tone periodic recovery With $n = 1000$ samples as previously, we set $\omega^* = 2$, $K = 25$ and $a_k = (-0.8)^k$ for $k < K$, to construct the more complicated periodic target signal $f^* : x \mapsto \sum_{k<K} a_k \cos((k+1)\,\omega^* x + \pi/12)$. The results are depicted in Figure 3.

Frequency estimation. In Figure 4, we compare the frequency estimation capability of our method with periodogram maximization (Kootsookos, 1993) for $n = 100$ and $n = 1000$ samples on a window of size $R = 10$, with independent identically distributed noise (normal

Proceedings Track

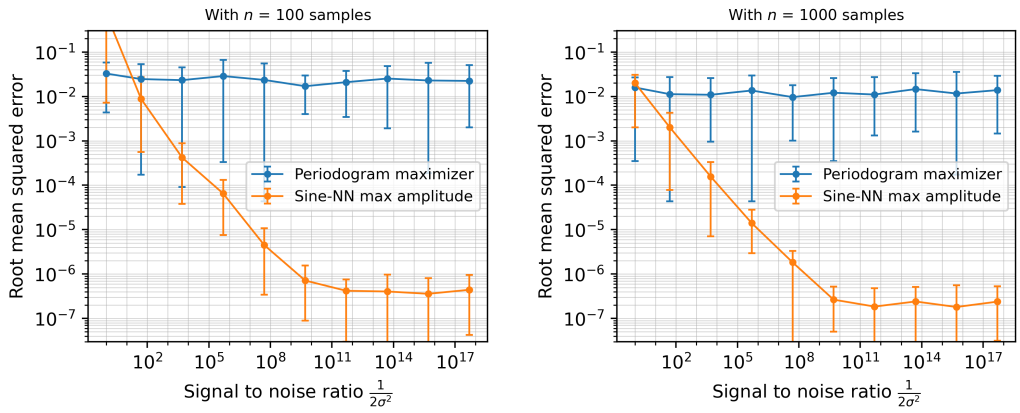

Figure 4: Frequency estimation performance with respect to noise level

distribution with mean zero and standard deviation σ for an amplitude of 1.0, yielding a signal-to-noise ratio of $1/2\sigma^2$). For our method, we estimate the whole signal in 10^5 iterations, then return the frequency ω_k associated with the maximum final amplitude $k = \text{argmax}_i(a_i^2 + b_i^2)$. Note that the computational cost of the algorithms are hardly comparable, ours requires solving an optimization problem every time, whereas periodogram maximization is nearly-instantaneous here. However this may still be useable in a regime where very few data points are available. We depict the mean root squared error over 15 independent runs with true frequency sampled uniformly in $[\omega_0, 15]$ for each run, where $\omega_0 = R/2\pi \approx 1.59$ is the minimal frequency to ensure that at least one period has been observed. Error bars depict the minimum and maximum value of the root squared error.

4. Conclusion

We have shown with simple and well-defined examples that neural networks with sine activations, often disregarded in favor of more complicated solutions resembling ReLU activations, require only minimal modifications to yield trainable architectures with competitive performance on the task of periodic signal recovery. While the direct applicability of this architecture to more realistic signals and partially-periodic datasets remains as future work, these experiments provide good sanity checks and baselines for architectures designed to learn periodic trends in larger settings in the future.

References

Andrew Barron. Barron, a.e.: Universal approximation bounds for superpositions of a sigmoidal function. ieee trans. on information theory 39, 930-945. *Information Theory, IEEE Transactions on*, 39:930 – 945, 06 1993. doi: 10.1109/18.256500.

George V. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.

Steffen Eger, Paul Youssef, and Iryna Gurevych. Is it time to swish? comparing deep learning activation functions across NLP tasks. *CoRR*, abs/1901.02671, 2019. URL http://arxiv.org/abs/1901.02671.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/gunasekar18a.html.

Boris Hanin and David Rolnick. Complexity of linear regions in deep networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2596–2604. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/hanin19a.html.

Peter Kootsookos. A review of the frequency estimation and tracking problems. 01 1993.

Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(05)80131-5. URL https://www.sciencedirect.com/science/article/pii/S0893608005801315.

Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. *CoRR*, abs/2104.03960, 2021. URL https://arxiv.org/abs/2104.03960.

Giambattista Parascandolo, Heikki Huttunen, and Tuomas Virtanen. Taming the waves: sine as activation function in deep neural networks, 2017. URL https://openreview.net/forum?id=Sks3zF9eg.

Liu Ziyin, Tilman Hartwig, and Masahito Ueda. Neural networks fail to learn periodic functions and how to fix it. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1583–1594. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1160453108d3e537255e9f7b931f4e90-Paper.pdf.