Supplementary Material for: From Activation to Initialization: Scaling Insights for Optimizing Neural Fields

Hemanth Saratchandran*
Australian Institute of Machine Learning,
University of Adelaide, Australia

Sameera Ramasinghe
Amazon, Australia

Simon Lucey
Australian Institute of Machine Learning,
University of Adelaide, Australia

1. Theoretical Analysis and Results

1.1. Notation

We will be using the same notation as **sec. 3** in the main paper. We review this notation for the convenience of the reader.

We consider a depth L neural network with layer widths $\{n_1, \dots, n_L\}$. We let $X \in \mathbb{R}^{N \times n_0}$ denote the training data, with n_0 being the dimension of the input and N being the number of training samples. We let $Y \in \mathbb{R}^{N \times n_L}$ denote the training labels. The output at layer k will be denoted by F_k and is defined by

$$F_k = \begin{cases} F_{L-1}W_L + b_L, & k = L \\ \phi(F_{k-1}W_k + b_k), & k \in [L-1] \\ X, & k = 0 \end{cases} \tag{1.1}$$

where the weights $W_k \in \mathbb{R}^{n_{k-1} \times n_k}$ and the biases $b_k \in \mathbb{R}^{n_k}$ and ϕ is an activation applied component wise. The notation $[m]$ is defined by $[m] = \{1, \dots, m\}$. For a weight matrix W_k at layer k, the notation W_k^0 will denote the initialization of that weight matrix. These are the initial weights of the network before training begins under a gradient descent algorithm. In general, we will denote the whole neural network by F. A neural field is any such network that parameterizes a continuous field.

All networks will be trained with the standard Mean Square Error (MSE) loss defined by

$$\mathcal{L}(\theta) = \frac{1}{2}||F_L(\theta) - Y||_F^2 \tag{1.2}$$

where θ denotes the parameters of the network F_L i.e. the weights and biases.

For the activation ϕ we will be primarily focused on the following functions:

1. Gaussian activations [2, 15, 18]: e^{-x^2/ω^2} where $\omega > 0$ is a fixed hyperparameter. The square of this hyperparameter is the variance of the Gaussian i.e. how wide the Gaussian is.
2. Sine activations [19]: $sin(\omega x)$ where $\omega > 0$ is a fixed hyperparameter, which gives the frequency of the sine function.
3. Sinc activations [16]: $sinc(\omega x) := \frac{sin(\omega x)}{\omega x}$ for $x \neq 0$ and 1 for $x = 0$.
4. Wavelet [17]: $e^{i\omega_0 x}e^{-x^2/\omega_1^2}$, where i denotes the complex imaginary number, $\omega_0 > 0$ is a frequency hyperparameter and $\omega_1^2 > 0$ is a variance hyperparameter. Observe that this activation is complex valued, hence when dealing with real-valued

*hemanth.saratchandran@adelaide.edu.au

signals we will always take the real part of the associated output of the neural network. In fact we will always deal with real-valued signals and hence will never need to consider the imaginary part. Note sometimes this wavelet is explicitly called the Gabor wavelet. When the context is clear we shall just call this activation the wavelet activation.

5. ReLU: This is the Rectified Linear Unit that is commonly used throughout all of machine learning.

For this section we will generally assume the frequency and variance parameters are fixed. In the experiments sections, sec. 2 and 3, we will choose these parameters by runnnin sweeps and picking the best values.

When we employ a $ReLU$ activation we will primarily do so with a positional embedding layer (PE), which is simply an embedding of the data into a higher dimensional space. The reader who is unfamiliar with PE should consult the standard references [10, 20]. A $ReLU$-networks that employs a positional embedding layer will be denoted by $ReLU - PE$.

We will also need the following complexity notations: $\mathcal{O}(\cdot)$, $\Theta(\cdot)$ and $\Omega(\cdot)$. We recall that $f(x) = \mathcal{O}(g(x))$ if there exists constants $C > 0$ and $x_0 > 0$ such that $f(x) \leq Cg(x)$ for all $x \geq x_0$. We have that $f(x) = \Omega(g(x))$ if the reverse inequality to to big O is true. Namely, that there exists constants $C > 0$ and $x_0 > 0$ such that $0 \leq Cg(x) \leq f(x)$ for all $x \geq x_0$. Finally, we remind that reader that $f(x) = \Theta(g(x))$ if $f(x)$ satisfies both sides of the inequality, that is $f(x) = \mathcal{O}(g(x))$ and $f(x) = \Omega(g(x))$.

Finally, we use the notation **w.h.p** to denote *with high probability* and the notation **w.p.** to denote *with probability/*.

1.2. Conditions for the convergence of gradient descent

In this section we will prove a preliminary result that will be integral to the proofs of the main theorems in **sec. 4** of the main paper. F will denote a fixed network of depth L as given by (1.1). When we want to emphasize that the network is of depth L, we will denote the network by F_L. We will denote the gradient descent updates by

$$\theta_{k+1} = \theta_k - \eta \nabla_k \mathcal{L}(\theta_k) \tag{1.3}$$

and let $F_l^k = F_l(\theta_k)$, where θ_k denotes the parameters at iteration k of the gradient descent update. When context clearly implies the iteration step, we shall simply write F_l and forget the k-superscript. Furthermore, when we want to talk about the weight matrices at a particular gradient descent iteration k, we will denote them by $(W_l^k)_{l=1}^L$. Thus superscripts will always denote the iteration step of the gradient descent algorithm.

In this section, we will assume the activation ϕ satisfies the following two inequalities

$$|\phi(x)| \leq A \text{ and } |\phi'(x)| \leq B. \tag{1.4}$$

Example 1.1. *Examples of activation functions satisfying eqns. (1.4) are sine, Gaussian, sinc, wavelet, tanh and sigmoid.*

Example 1.2. *Examples of activation functions not satisfying eqns. (1.4) are ReLU, GeLU and SiLU.*

The following lemma bounds the kth-layer feature matrix for $0 \leq k \leq L-1$ in terms of a bound on the activation, the width of the layer and the number of samples.

Lemma 1.3. *Let ϕ be an activation function such that $|\phi| \leq A$. Then for $0 \leq k \leq L-1$, we have that*

$$||F_k||_2 \leq ||F_k||_F \leq \sqrt{An_kN}$$

Proof. For a give matrix the 2-norm $||\cdot||_2$ is given by the maximum singular value of the matrix, which is always bounded above by the Frobenius norm. To obtain the bound on the Frobenius norm, observe that each entry of F_k is given by $(F_k)_{ij} = \phi((W_kF_{k-1})_{ij})$, since the activation function act component wise. In particular, each entry of F_k is bounded above by A. Since there are a total of n_kN entries the result follows. □

The following lemma provides a bound on the gradient of the loss.

Lemma 1.4. *For $\theta = (W_p)_{p=1}^L$ and $k \in [L]$ we have*

$$\begin{aligned} &||\nabla_{W_k} \mathcal{L}(\theta)||_F \\ &\leq B^{L-(k+1)} \sqrt{An_{k-1}N} \prod_{p=k+1}^{L} ||W_p||_2 ||F_L - Y||_2 \end{aligned}$$

where we remind the reader that N is the number of training samples.

Proof. The lemma is proved using the formula

$$Vec\Big(\nabla_{W_k}\mathcal{L}(\theta)\Big) = \big(F_L(\theta) - Y\big)\prod_{p=k+1}^{L}\big(W_p \otimes I_N\big)D_{p-1}\cdot I_{n_l}\otimes F_{k-1},$$

where Vec denotes the vectorization operator. See [11] for a derivation of the formula.

We then estimate

$$\begin{aligned}
||\nabla_{W_k}\mathcal{L}(\theta)||_F &= ||Vec\Big(\nabla_{W_k}\mathcal{L}(\theta)\Big)||_2 \\
&= \Big|\Big|\big(F_L(\theta) - Y\big)\prod_{p=k+1}^{L}\big(W_p \otimes I_N\big)D_{p-1}\cdot I_{n_l}\otimes F_{k-1}\Big|\Big|_2 \\
&\leq B^{L-(k+1)}||F_L(\theta) - Y||_2||F_{k-1}||_2\prod_{p=k+1}^{L}||W_p||_2 \\
&\leq B^{L-(k+1)}\sqrt{An_{k-1}N}\prod_{p=k+1}^{L}||W_p||_2||F_L(\theta) - Y||_2
\end{aligned}$$

where the last inequality comes from applying lemma 1.3. □

The following lemma bounds the difference of a feature matrix at two different iteration points, along gradient descent.

Lemma 1.5. *Let $\theta_a \in (W_l^a)_{l=1}^L$, $\theta_a \in (W_l^b)_{l=1}^L$ and choose numbers $\overline{\lambda}_l \geq \max\{||W_l^a||_2, ||W_l^b||_2\}$. Then for any $k \in [L]$ we have*

$$\begin{aligned}
&||F_k(\theta_a) - F_k(\theta_b)||_F \\
&\leq B^k||X||_F\Big(\prod_{j=1}^{k}\overline{\lambda}_j\Big)||W_1^a - W_1^b||_F \\
&\quad + \sum_{i=2}^{k}B^{k-(i-1)}\sqrt{An_{i-1}N}\Big(\prod_{j=1}^{k}\overline{\lambda}_j\Big)||W_i^a - W_i^b||_F
\end{aligned}$$

Proof. The proof proceeds via induction. We start by proving the base $k = 1$ case

$$\begin{aligned}
||F_1^a - F_1^b||_F &= ||\sigma(W_1^aX) - \sigma(W_1^bX)||_F \\
&\leq B||X||_F||W_1^a - W_1^b||_F.
\end{aligned}$$

Assume the statement for $k-1$, we then prove it for k:

$$\begin{aligned}
||F_k^a - F_k^b||_F &= ||\sigma(W_k^aF_{k-1}^a) - \sigma(W_k^bF_{k-1}^b)||_F \\
&\leq B||W_k^aF_{k-1}^a - W_k^bF_{k-1}^b||_F \\
&\leq B||W_k^aF_{k-1}^a - W_k^aF_{k-1}^b + W_k^aF_{k-1}^b - W_k^bF_{k-1}^b||_F \\
&\leq B\big(||W_k^aF_{k-1}^a - W_k^aF_{k-1}^b||_F + ||W_k^aF_{k-1}^b - W_k^bF_{k-1}^b||_F\big) \\
&\leq B\big(||W_k^a||_2||F_{k-1}^a - F_{k-1}^b||_F + ||F_{k-1}^b||_2||W_k^a - W_k^b||_F\big) \\
&\leq B||W_k^a||_2||F_{k-1}^a - F_{k-1}^b||_F + B\sqrt{An_{k-1}N}||W_k^a - W_k^b||_F
\end{aligned}$$

where the last inequality follows from applying lemma 1.3. Applying the induction hypothesis to the $||F_{k-1}^a - F_{k-1}^b||_F$ term proves the lemma. □

We now prove a theorem that gives conditions under which gradient descent converges to a global minimum. It will be used to prove the main theorems in **sec. 4** of the main paper by showing that under certain initializations the conditions of the theorem are satisfied, thus allowing use to deduce that such initializations lead to convergence to a global minimum.

Theorem 1.6. *Fix a deep neural network F given by (1.1), having activation function ϕ satisfying (1.4) and such that $n_{L-1} \geq N$. Let $(C_l)_{l=1}^{L}$ be any fixed sequence of positive numbers and let*

$$\sigma_0 = \sigma_{min}(F_{L-1}^0),\ \overline{\lambda}_l = ||W_l^0||_2 + C_l,\ \overline{\lambda}_{i\to j} = \prod_{l=i}^{j} \overline{\lambda}_l, \tag{1.5}$$

where $\sigma_{\min}$ denotes the minimum singular value. Assume that the following inequalities are satisfied at initialisation:

$$\sigma_0^2 \geq 16\sqrt{2\mathcal{L}(\theta)}(\overline{B})^{L-2}\overline{A}\left(\frac{\max_{l\in[L]}\{\overline{\lambda}_{l+1\to L}\}}{\min_{l\in[L]}\{C_l\}}\right) \tag{1.6}$$

$$\sigma_0^3 \geq 32\sqrt{2\mathcal{L}(\theta_0)}\left(\overline{B}^{2L-2}||X||_F(\sqrt{An_0N})\overline{\lambda}_{2\to L}\prod_{j=1}^{L-1}\overline{\lambda}_j + \sum_{i=2}^{L-1}\overline{B}^{2(L-i)-1}(An_{i-1}N)\overline{\lambda}_{i+1\to L}\prod_{j=i+1}^{L-1}\overline{\lambda}_j\right) \tag{1.7}$$

$$\sigma_0^2 \geq 8\overline{\lambda}_L\left(\overline{B}^{2L}||X||_F\sqrt{An_oN}(\overline{\lambda_1})(\overline{\lambda}_{2\to L-1})^2 + \sum_{i=2}^{L-1}\overline{B}^{2L}(An_{i-1}N)(\overline{\lambda}_{i+1\to L-1})^2\right)\overline{\lambda}_L \tag{1.8}$$

where $\overline{B} = \max\{1, B\}$ and $\overline{A} = \max_{l\in[L]}\{\sqrt{An_lN}\}$.

Assume that the learning rate satisfies

$$\eta < \min\left\{\frac{4}{\sigma_0^2}, (\overline{\lambda}_L)^{-3}\left(B^{2L}||X||_F\sqrt{An_oN}(\overline{\lambda_1})(\overline{\lambda}_{2\to L-1})^2 + \sum_{i=2}^{L}B^{2L}(An_{i-1}N)(\overline{\lambda}_{i+1\to L-1})^2\right)^{-1}\right\}. \tag{1.9}$$

Then for any $k \geq 0$ we have that

$$\mathcal{L}(\theta_t) \leq \left(1 - \eta\frac{\sigma_0^2}{4}\right)^k \mathcal{L}(\theta_0) \tag{1.10}$$

implying that the loss function $\mathcal{L}$ converges to a global minimum under gradient descent.

Proof. We will prove that for every $k \geq 0$ the following conditions are satisfied:

$$||W_l^t||_2 \leq \overline{\lambda}_l,\ l \in [L],\ t \in [0, k], \tag{1.11}$$

$$\frac{1}{2}\sigma_0 \leq \sigma_{min}(F_{L-1}^t),\ t \in [0, k], \tag{1.12}$$

$$\mathcal{L}(\theta_t) \leq \left(1 - \eta\frac{\sigma_0^2}{8}\right)^t \mathcal{L}(\theta_0),\ t \in [0, k]. \tag{1.13}$$

The proof will proceed via induction with the base $k = 0$ being clear. Therefore, assume that the three conditions hold for iteration k. We will prove they hold for iteration $k+1$. Applying the triangle inequality and using the gradient decent update,

we have

$$
\begin{aligned}
&||W_l^{k+1} - W_l^0||_F \\
&\leq \sum_{s=0}^{k} ||W_l^{s+1} - W_l^s||_F \\
&= \eta \sum_{s=0}^{k} ||\nabla_{W_l} \mathcal{L}(\theta_s)||_F \\
&\leq \eta \sum_{s=0}^{k} B^{L-(l+1)} \sqrt{An_{l-1}N} \prod_{p=l+1}^{L} ||W_p^s||_2 ||F_L^s - Y||_2 \\
&\leq \eta B^{L-(l+1)} \sqrt{An_{l-1}N} \sum_{s=0}^{k} \overline{\lambda}_{l+1\to L} ||F_L^s - Y||_2 \\
&\leq \eta B^{L-(l+1)} \sqrt{An_{l-1}N} (\overline{\lambda}_{l+1\to L}) \cdot \sum_{s=0}^{k} \left(1 - \eta \frac{\sigma_0^2}{8}\right)^{s/2} ||F_L^0 - Y||_F
\end{aligned}
$$

where the second inequality follows from lemma 1.4, and the final inequality from the induction assumption. Let $u = \left(1 - \eta \frac{\sigma_0^2}{8}\right)^{1/2}$. The above right term can then be bounded by

$$
\begin{aligned}
&\eta B^{L-(l+1)} \sqrt{An_{l-1}N} (\overline{\lambda}_{l+1\to L}) \sum_{s=0}^{k} u^s ||F_L^0 - Y||_F \\
&\leq \frac{8}{\sigma_0^2} B^{L-(l+1)} \sqrt{An_{l-1}N} (\overline{\lambda}_{l+1\to L}) \cdot (1 - u^2) \frac{(1 - u^{k+1})}{1 - u} ||F_L^0 - Y||_F \\
&\leq \frac{16}{\sigma_0^2} B^{L-(l+1)} \sqrt{An_{l-1}N} (\overline{\lambda}_{l+1\to L}) ||F_L^0 - Y||_F \qquad (1.14) \\
&\leq C_l, \text{ by (1.6)}
\end{aligned}
$$

where we used that $0 < u < 1$ to get the second inequality. Using the fact that the operator norm is bounded above by the Frobenius norm and an application of Weyl's inequality gives

$$
||W_l^{k+1}||_2 \leq ||W_l^0||_2 + C_l = \overline{\lambda}_l \qquad (1.15)
$$

and thus condition (1.11) has been proved for $k + 1$.

We move on to prove condition (1.12). We have

$$
\begin{aligned}
&||F_{L-1}^{k+1} - F_{L-1}^0||_F \\
&\leq B^L ||X||_F \left(\prod_{j=1}^{L-1} \overline{\lambda}_j \right) ||W_1^{k+1} - W_1^0||_F + \sum_{i=0}^{L-1} B^{(L-1)-(i-1)} \sqrt{An_{l-1}N} \left(\prod_{j=i+1}^{L-1} \overline{\lambda}_j \right) \cdot ||W_i^{k+1} - W_i^0||_F \\
&\leq \left[\left(\frac{16}{\sigma_0^2} \right) B^{2L-2} ||X||_F \left(\sqrt{An_0 N} \right) \left(\prod_{j=1}^{L-1} \overline{\lambda}_j \right) \overline{\lambda}_{2\to L} + \sum_{i=0}^{L-1} B^{2(L-i)-1} \sqrt{An_{i-1}N} \left(\prod_{j=i+1}^{L-1} \overline{\lambda}_j \right) \left(\overline{\lambda}_{i+1\to L} \right) \right] \sqrt{\mathcal{L}(\theta_0)} \\
&\leq \frac{1}{2} \sigma_0, \text{ by (1.7)}
\end{aligned}
$$

where the first inequality follows from lemma 1.5 and the second by applying (1.14). This establishes condition (1.13) for $k + 1$. The final step is to prove (1.13).

Define the matrix $G = F_{L-1}^k W_L^{k+1}$. A simple computation shows

$$
\begin{aligned}
2\mathcal{L}(\theta_{k+1}) &= ||F_L^{k+1}(\theta) - Y||_F^2 \\
&= 2\mathcal{L}(\theta_k) + ||F_L^{k+1} - F_L^k||_F^2 + 2Tr(F_L^{k+1} - F_L^k)(F_L^k - Y)^T \\
&= 2\mathcal{L}(\theta_k) + ||F_L^{k+1} - F_L^k||_F^2 + 2Tr(F_L^{k+1} - G)(F_L^k - Y)^T + 2Tr(G - F_L^k)(F_L^k - Y)^T.
\end{aligned}
$$

The strategy now is to bound each term on the right separately, put them together and obtain a bound for the left hand side of the above inequality. Using lemmas 1.4 and 1.5, we have

$$\begin{aligned}||F_L^{k+1} - F_L^k||_F &\leq B^L||X||_F\left(\prod_{j=1}^{L}\overline{\lambda}_j\right)||W_l^{k+1} - W_l^k||_F + \sum_{i=2}^{L} B^{L-(i-1)}\left(\sqrt{An_{i-1}N}\right)\left(\prod_{j=i+1}^{L}\overline{\lambda}_j\right)||W_i^{k+1} - W_i^k||_F \\ &\leq \eta\left[B^{2L-2}||X||_F\left(\sqrt{An_0N}\right)\overline{\lambda}_{1\to L}\overline{\lambda}_{2\to L} + \sum_{i=2}^{L} B^{2(L-i)}(An_{i-1}N)\left(\overline{\lambda}_{i+1}\right)^2\right]||F_L^k - Y||_F.\end{aligned}$$

We then observe that by applying lemma 1.5 and (1.15) we get

$$\begin{aligned}Tr(F_L^{k+1} - G)(F_L^k - Y)^T &= Tr(W_L^{k+1}F_{L-1}^{k+1} - W_L^{k+1}F_{L-1}^k)(F_L^k - Y)^T \\ &\leq ||F_{L-1}^{k+1} - F_{L-1}^k||_F||W_L^{k+1}||_2||F_L^k - Y||_F \\ &\leq \eta\left[B^{2(L-1)}||X||_F\left(\sqrt{An_0N}\right)\overline{\lambda}_{1\to L-1}\overline{\lambda}_{2\to L-1} + \sum_{i=2}^{L-1} B^{2(L-1-i)}(An_{i-1}N)\left(\overline{\lambda}_{i+1\to L-1}\right)^2\right] \\ &\quad \cdot \overline{\lambda}_L||F_L^k - Y||_F^2.\end{aligned}$$

The final step is to estimate the term $Tr(G - F_L^k)(F_L^k - Y)^T$:

$$\begin{aligned}Tr(G - F_L^k)(F_L^k - Y)^T &= -\eta Tr\left((F_{L-1}^k)^T(F_{L-1}^k)(F_L^k - Y)^T(F_L^k - Y)\right) \\ &\leq -\eta\sigma_{\min}(F_{L-1}^k)^2||F_L^k - Y||_F^2 \\ &\leq -\eta\frac{\sigma_0^2}{4}||F_L^k - Y||_F^2\end{aligned}$$

where we used the fact that

$$\nabla_{W_L}\mathcal{L}(\theta_k) = (F_{L-1}^k)^T(F_L^k - Y),$$

our assumption that $n_{L-1} \geq N$ to obtain $\lambda_{\min}((F_{L-1}^k)^T(F_{L-1}^k)) = \sigma_{\min}(F_{L-1}^k)^2$, and the induction hypothesis for k giving, $\sigma_{\min}(F_{L-1}^k) \geq \frac{1}{2}\sigma_0$.

We can now form an estimate for $\mathcal{L}(\theta_{k+1})$ that will prove condition (1.13). We define two terms

$$\begin{aligned}T_1 &= \left[\overline{B}^{2L}||X||_F\left(\sqrt{An_0N}\right)(\overline{\lambda}_1)(\overline{\lambda}_{2\to L-1})^2 + \sum_{i=2}^{L}\overline{B}^{2L}(An_{i-1}N)\left(\overline{\lambda}_{i+1\to L-1}\right)^2\right](\overline{\lambda}_L)^2 \\ T_2 &= \left[\overline{B}^{2L}||X||_F\left(\sqrt{An_0N}\right)(\overline{\lambda}_1)(\overline{\lambda}_{2\to L-1})^2 + \sum_{i=2}^{L-1}\overline{B}^{2L}(An_{i-1}N)\left(\overline{\lambda}_{i+1\to L-1}\right)^2\right]\overline{\lambda}_L\end{aligned}$$

and note that condition (1.13) says precisely that $\sigma_0^2 \geq 8T_2$. Using this we obtain

$$\begin{aligned}\mathcal{L}(\theta_{k+1}) &\leq \left(1 - \eta\frac{\sigma_0^2}{2} + 2\eta T_2\right)\mathcal{L}(\theta_k) \\ &\leq \left(1 - \eta\frac{\sigma_0^2}{4}\right)\mathcal{L}(\theta_k).\end{aligned}$$

Finally, the assumption (1.9) on η finishes the induction proof. □

The above theorem gives three conditions that need to be satisfied for the gradient descent algorithm to converge for the MSE loss function of a neural network F admitting an activation function ϕ that satisfies eqns. (1.4). In particular these three conditions can be used to check if a neural network admitting one of the groups of activation functions given in exmp. 1.1.

1.3. Do the inequalities (1.6)-(1.8) from theorem 1.6 actually hold?

The purpose of this subsection is to show that there are many regions in parameter space where conditions (1.6)-(1.8) hold. Theorem 2.2 then guarantees that if a network is initialised at such points of parameter space, then with a small enough learning rate, gradient decent will converge to a global minimum.

Set $C_l = 1$ for $l \in [L]$. Initialise the weights $\theta_0 = W_1^0, \dots, W_{L-1}^0, W_L^0$ so that $\sigma_{min}(F_{L-1}(\theta_0)) > 0$ and $W_L^0 = 0$. This latter condition implies $\sqrt{2\mathcal{L}(\theta_0)} = ||Y||_F$. Let $\theta_0(r) = (rW_1^0, \dots, rW_{L-1}^0, 0)$ be an r dependent parameter.

Condition (1.6) for $\theta_0(r)$ reads

$$\sigma_0(\theta_0(r))^2 \geq 16\sqrt{2\mathcal{L}(\theta)}(\overline{B})^L \overline{A}\big(\max_{l\in[L]}\{\overline{\lambda}_{l\to L}(\theta_0(r)\}\big). \tag{1.16}$$

The term

$$\overline{\lambda}_{l\to L}(\theta_0(r)) = \prod_{k=l}^{L-1}\big(r||W_l^0|| + 1\big)$$

is a polynomial of degree at most $L-1$ in r. Furthermore, since $\sigma_0(\theta_0(r)) = \sigma_{min}(F_{L-1}(\theta_0(r))$ it follows that $\sigma_0(\theta_0(r))^2 = r^{2(L-1)}\sigma_0^2$. We thus see that the left hand side of inequality (1.16) is a polynomial of degree $2(L-1)$ in r and the right hand side is a polynomial of degree at most $L-1$ in r. It thus follows that for r sufficiently large (1.16) holds. We can analyse condition (1.7) in a similar way. Substituting $\theta_0(r)$ into (1.7), gives the following inequality for r

$$\frac{r^{3(L-1)}}{32} \geq \sqrt{2\mathcal{L}(\theta_0)}\Bigg(r^{2L-3}\overline{B}^{2L-2}||X||_F(\sqrt{An_0N})\overline{\lambda}_{2\to L}\overline{\lambda}_{1\to L-1} + \sum_{i=2}^{L-1} r^{2(L-1-i)}\overline{B}^{2(L-i)-1}(An_{i-1}N)\overline{\lambda}_{i+1\to L}\prod_{j=1}^{L-1}\overline{\lambda}_j\Bigg).$$

The term on the left is a degree $3(L-1)$ polynomial in r and the term on the right has degree at most $2L-3$. This shows the above inequality can be satisfied for r sufficiently large. A similar analysis for equation (1.8) shows that it too can be satisfied for r sufficiently large. In particular, we see that the last hidden layer need only have width N and provided the network is initialised at such a point, with r sufficiently large, thm. 1.6 guarantees convergence to a global minimum, for a small enough learning rate. The above argument also shows there are parameters for which conditions (1.6)-(1.8) will not hold. Thus we see that initialisation is crucial to being able to apply thm. 1.6.

1.4. Preliminary lemmas on norms of the feature matrices: Part 1

In this section we will obtain some important norm bounds on the feature matrices of a neural network F as defined in (1.1). The results of this section are only valid for the following four activations:

1. Gaussian
2. sine
3. sinc
4. wavelet

see sec. 1.1 for their definition. Since the wavelet activation is given by $e^{i\omega_0 x}e^{-x^2/\omega_1^2}$ and hence $|e^{i\omega_0 x}e^{-x^2/\omega_1^2}| = e^{-x^2/\omega_1^2}$, we find that the same proof for the Gaussian activation works for the wavelet activation. Furthermore, since the norm of the sinc activation is bounded above by 1 for $|x| \leq 1$ and by $sin(x)$ for $|x| > 1$, we find that the same proof that is given for the sine activation allows goes through for the sinc activation. Therefore, we will primarily focus on the sine and Gaussian activations for this section.

We will need to make use of the sub-exponential and sub-Gaussian norms, which we now describe. Given a sub-exponential random variable X, define

$$||X||_{\psi_1} = \inf\{t > 0 : \mathbb{E}[exp(|X|/t)] \leq 2\}.$$

Given a sub-Gaussian random variable X define the sub-Gaussian norm

$$||X||_{\psi_2} = \inf\{t > 0 : \mathbb{E}[exp(|X^2|/t^2)] \leq 2\}.$$

Lemma 1.7. *Let ϕ be the activation given by $sin(\omega x)$. Then $||\phi(Xw)||_{\psi_2} = \mathcal{O}(||X||_F)$.*

Proof. We need to look at the integral

$$\int_{\mathbb{R}^{n_0}} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg). \tag{1.17}$$

Observe that $|sin(\omega x)| \leq \omega|x|$. We can then bound the integral in the following way

$$\int_{\mathbb{R}^{n_0}} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg) \leq \int_{\mathbb{R}^{n_0}} exp\bigg(\omega^2\frac{||X||_F^2||w||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg).$$

Let $t^2 = m||X||_F^2\beta$ for some $m > 0$ to be chosen later.

Then we get that the above can be written as

$$\begin{aligned}\int_{\mathbb{R}^{n_0}} exp\bigg(\frac{||w||_2^2}{m\beta^2}\bigg)exp\bigg(\frac{||w||^2}{\beta^2}\bigg)dw &= \int_{\mathbb{R}^{n_0}} exp\bigg(-\bigg(1-\frac{1}{m}\bigg)\frac{||w||^2}{\beta^2}\bigg)dw \\ &\leq \bigg(\frac{m}{m-1}\bigg)^{\frac{n_0}{2}}\beta^{n_0} \\ &< 2, \text{ for } m \text{ large.}\end{aligned}$$

The lemma follows. □

Lemma 1.8. *Let ϕ be the activation given by $exp(-x^2/\omega^2)$. Then $||\phi(Xw)||_{\psi_2} = \mathcal{O}(||X||_F)$ w.p 1.*

Proof. We need to look at the integral

$$\int_{\mathbb{R}^{n_0}} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg). \tag{1.18}$$

Choose $\epsilon > 0$ so that $|\exp(-x^2/\omega^2)| \leq d(\epsilon)|x|$, where $d(\epsilon) > 0$, on $\mathbb{R}\backslash(-\epsilon,\epsilon)$. Define $\chi(\epsilon) = \{w \in \mathbb{R}^{n_0} : Xw \in [-\epsilon,\epsilon]^N\}$, i.e. $\chi(\epsilon)$ denotes the set of those vectors that are mapped into a small cube about the origin in $\mathbb{R}^N$ by X, viewed as a matrix.

We then compute the integral in the following way.

$$\begin{aligned}\int_{\mathbb{R}^{n_0}} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg) &= \int_{\mathbb{R}^{n_0}\backslash\chi(\epsilon)} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg) \\ &+ \int_{\chi(\epsilon)} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg).\end{aligned}$$

The first integral can be bounded above by 2 taking $t^2 = m||X||_F^2\beta$ and applying the same argument as in lemma 1.7. The second integral can be estimated as follows

$$\begin{aligned}\int_{\chi(\epsilon)} exp\bigg(\frac{||\phi(Xw)||_2^2}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg) &\leq \int_{\chi(\epsilon)} exp\bigg(\frac{Nn_0}{t^2}\bigg)exp\bigg(\frac{-||w||^2}{\beta^2}dw\bigg) \\ &= exp\bigg(\frac{Nn_0}{t^2}\bigg)\int_{\chi(\epsilon)} exp\bigg(\frac{-||w||^2}{\beta^2}\bigg)dw \\ &\leq exp\bigg(\frac{Nn_0}{t^2}\bigg)\int_{\chi(\epsilon)} dw \\ &\to exp\bigg(\frac{Nn_0}{t^2}\bigg)\int_{\chi(0)} dw \\ &= 0, \, w.h.p\end{aligned}$$

Note that $\chi(0) = Ker(X)$ which has measure 0 in $\mathbb{R}^{n_0}$ w.p 1. The lemma follows.

□

Proposition 1.9. *Let $\delta > 0$, then $\sigma_0^2 \geq \frac{n_1\lambda}{4}$ w.p $1-\delta$ if if $n_1 = \widetilde{\Omega}(N/\lambda)$, where $\lambda = \lambda_{\min}(G)$ and*

$$G = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\right) \tag{1.19}$$

Proof. Let $A \in \mathbb{R}^{N\times n_1}$ be a random Gaussian matrix such that $A_{:j} = \phi(XW_{:j})\mathbf{1}_{||\phi(XW_{:j})||_2\leq t}$, where $\mathbf{1}_{||\phi(XW_{:j})||_2\leq t}$ denotes a characteristic function and $t = ||X||_F \max\{1, log\frac{2||X||_F^2}{\lambda}\}$, where we define λ shortly. Let

$$G = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\right)$$
$$G^* = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\mathbf{1}_{||\phi(XW)||_2\leq t}\right).$$

Then $\lambda_{min}(F_kF_k^T) \geq \lambda_{\min}(AA^T)$ and $\lambda_{\max}(A_{:j}A_{:j}^T) \leq t^2$. We can then apply the matrix Chernoff inequality, see theorem 1.1 [21], to obtain that for any $\epsilon \in [0,1)$

$$\mathbb{P}(\lambda_{min}(AA^T) \leq (1-\epsilon)\lambda_{\min}(\mathbb{E}AA^T)) \leq N\left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{\lambda_{min}(\mathbb{E}AA^T)/t^2}. \tag{1.20}$$

Taking $\epsilon = 1/2$, we find

$$\mathbb{P}\left(\lambda_{min}(AA^T) \leq n_1\lambda_{\min}(G^*)/2\right) \leq exp(-cn_1\lambda_{\min}(G^*)/t^2 + Log(N)) \tag{1.21}$$

Thus for $n_1 \geq \frac{t^2}{c\lambda_{min}(G^*)}Log(N/\delta)$ we have $\lambda_{\min}(AA^T) \geq \frac{n_1\lambda_{\min}(G^*)}{2}$ w.p $\geq 1-\delta$.
We then find

$$\begin{aligned}
||G^* - G||_2 &\leq \mathbb{E}\left(\phi(XW)\phi(XW)^T\mathbf{1}_{||\phi(XW)||_2\leq t} - \phi(XW)\phi(XW)^T||_2\right)\\
&= \mathbb{E}\left(||\phi(XW)||_2^2\mathbf{1}_{||\phi(XW)||_2>t}\right)\\
&= \int_{s=0}^{\infty}\mathbb{P}\left(||\phi(XW)||_2^2\mathbf{1}_{||\phi(XW)||_2>t} > \sqrt{s}\right)ds\\
&= \int_{s=0}^{\infty}\mathbb{P}\left(||\phi(XW)||_2^2 > t\right)\mathbb{P}\left(||\phi(XW)||_2^2 > \sqrt{s}\right)ds\\
&\leq \int_{s=0}^{\infty}\exp(-c\frac{t^2+s}{||\phi(XW)||_{\psi_2}^2})ds\\
&\leq \int_{s=0}^{\infty}\exp\left(-c\frac{t^2+s}{C||X||_F^2}\right)ds\\
&\leq \frac{\lambda}{2}
\end{aligned}$$

where the second inequality uses lemmas 1.7, 1.8.

It follows that $\lambda_{\min}(G^*) \geq \lambda/2$. Therefore, taking $n_1 \geq \max\{N, \frac{2t^2}{c\lambda}Log\frac{N}{\delta}\}$, it holds w.p $1-\delta$ that

$$\sigma_{\min}(F_1)^2 = \lambda_{min}(F_1F_1^T) \geq \lambda_{min}(AA^T) \geq n_1\lambda_{\min}(G^*)/2 \geq \frac{n_1\lambda}{4}.$$

□

Lemma 1.10. *Suppose ϕ is an activation function that satisfies $|\phi| \leq A$. For $l \in [L]$, let $(W_l)_{ij} \sim \mathcal{N}(0, \frac{C}{\beta})$. Then we have*

$$\mathbb{E}||F_l||_F^2 \leq \frac{\sqrt{C}An_lN}{\sqrt{\beta}}$$

Proof. $\mathbb{E}||F_l||_F^2 = \mathbb{E}||\phi(W_l F_{l-1})||_F^2 \leq An_l N\mathbb{E}1 = \frac{\sqrt{C}An_l N}{\sqrt{\beta}}$. □

Lemma 1.11. *Suppose ϕ is a C^1-differentiable function satisfying $|\phi| \leq A$ and $|\phi'| \leq B$. Then w.p $1 - Lexp(-t^2/2B^2)$ over $(W_l)_{l=1}^L \sim \mathcal{N}(0, 1/n_{l-1})$ we have*

$$||F_L||_F \leq \frac{\sqrt{An_L N}}{n_{L-1}^{1/4}} + \frac{\sqrt{An_{L-1}N}}{n_{L-2}^{1/4}n_{L-1}^{1/2}}t + \frac{\sqrt{An_{L-2}N}}{n_{L-3}^{1/4}n_{L-2}^{1/2}n_{L-3}^{1/2}}t^2 + \cdots + \frac{\sqrt{An_1 N}}{n_0^{1/4}n_1^{1/2}\cdots n_{L-1}^{1/2}}t^{L-1} \frac{||X||_F}{n_0^{1/2}\ldots n_{L-2}^{1/2}n_{L-1}^{1/2}}t^L$$

Proof. The proof is by induction over $l \in [L]$. We first prove the base $l = 1$ case. We note that each feature map F_l is Lipshitz from the assumptions on ϕ. Since W_1 is Gaussian distributed, we can apply Gaussian concentration, see [22], to obtain

$$||F_1||_F - \mathbb{E}||F_1||_F \sim subG\left(\frac{||X||_F^2 B^2}{n_0}\right)$$

which implies

$$\mathbb{P}\left(|||F_1||_F - \mathbb{E}||F_1||_F \leq \frac{||X||_F}{\sqrt{n_0}}t\right) \leq 1 - 2exp(-t^2/2B^2).$$

We therefore have that

$$\begin{aligned} ||F_1||_F &\leq \mathbb{E}||F_1||_F + \frac{||X||_F}{\sqrt{n_0}}t \\ &\leq \frac{\sqrt{An_1 N}}{n_0^{1/4}} + \frac{||X||_F}{\sqrt{n_0}}t \end{aligned}$$

where we have used lemma 1.10 to bound the expectation term.

We now assume the lemma is true for $l-1$, that is w.p $\geq 1 - (l-1)exp(-t^2/2B^2)$ we have that

$$||F_{L-1}||_F \leq \frac{\sqrt{An_{L-1}N}}{n_{L-2}^{1/4}} + \frac{\sqrt{An_{L-2}N}}{n_{L-3}^{1/4}n_{L-2}^{1/2}}t + \cdots + \frac{||X||_F}{n_0^{1/2}\ldots n_{L-2}^{1/2}n_{L-2}^{1/2}}t^{L-1}.$$

Conditioning on $(W_p)_{p=1}^{l-1}$ we have that F_l is Lipshitz and hence

$$||F_{l-1}||_F - \mathbb{E}||F_{l-1}||_F \sim subG\left(\frac{||F_{l-1}||_F^2 B^2}{n_{l-1}}\right)$$

implying

$$||F_l||_F \leq \mathbb{E}||F_l||_F + \frac{||F_{l-1}||_F}{\sqrt{n_{l-1}}}t$$

w.p $\geq 1 - exp(-t^2/2B^2)$ over W_l. The result then follows by applying the induction hypothesis and lemma 1.10. □

In the case of a Sinusoidal activation, we can prove much better bounds than lemmas 1.10, 1.11.

The proof of the following lemma follows from lemma C.3 in [13] by observing that $|sin(\omega x)| \leq \omega|x|$.

Lemma 1.12. *Let $\phi = sin(\omega x)$ and for each $l \in L$ let $(W_l)_{ij} \sim \mathcal{N}(0, \beta_l)$ for every $l \in [L]$. Then for each $l \in [L]$ we have that $\mathbb{E}||F_l||_F^2 \leq \omega^2 \beta_l n_l \mathbb{E}||F_{l-1}||_F^2$.*

Lemma 1.13. *Let $\phi = sin(\omega x)$ and for each $l \in L$ let $(W_l)_{ij} \sim \mathcal{N}(0, \beta_l)$ for every $l \in [L]$. Fix some $t > 0$ and assume that $\sqrt{n_l} \geq t$ for all $l \in [L]$. Then*

$$||F_L||_F \leq \prod_{l=1}^{L}\sqrt{n_l\beta_{l-1}} + L\left(\prod_{l=1}^{L-1}\sqrt{n_l\beta_l}\right)\beta_0||X||_F t$$

w.p $\geq 1 - Lexp(-t^2/2)$.

The proof of the above lemma follows in a similar way to lemma 1.11 using lemma 1.12.

1.5. Preliminary lemmas on norms of the feature matrices: Part 2

1.5.1 Preliminaries on Hermite polynomials and expansions

In this section we analyse the Hermite representations of the functions $sin(\omega x)$ and $e^{-\frac{x^2}{\omega^2}}$. As in sec. 1.4 we will focus on the case of sine and Gaussian as these results can then easily be extended to the case of sinc and wavelet. These representations will then be used to establish an estimate on the quantity $\lambda_{\min}(G)$, where $G = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\right)$.

We start with some background on Hermite expansions. We will consider the space $L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$ of weighted L^2 functions with inner product given by

$$\langle f, g\rangle = \int_{\mathbb{R}} f(x)g(x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx. \tag{1.22}$$

In other words, the vector space consists of equivalence classes of Lebesgue measurable functions on $\mathbb{R}$ that have finite weighted norm, given by equation (1.22).

The vector space $L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$ forms a Hilbert space with the above inner product. A well known orthonormal basis for this Hilbert space is given by the Hermite polynomials

$$h_n(x) = \frac{1}{\sqrt{n!}}(-1)^n e^{x^2/2}\frac{d^n}{dx^n}e^{-x^2/2} \tag{1.23}$$

for each $n \geq 0$.

Therefore, any function $f \in L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$ can be represented as an expansion of the form

$$f = \sum_{n=0}^{\infty} a_n h_n \tag{1.24}$$

where

$$a_n := \int_{\mathbb{R}} f(x)h_n(x)\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx. \tag{1.25}$$

The expansion given in (1.24) is known as the Hermite expansion of f. The above equation (1.24) is to be understood in the sense that the right hand side converges to the left hand side with respect to the norm of $L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$. Note that equation (1.24) implies that $f \in L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$ if and only if $\langle f, f\rangle = \sum_{n=0}^{\infty} a_n^2 < \infty$.

The Hermite polynomials satisfy various properties that are important in their analysis. We list a few that will be important for our analysis of Sinusoidal and Gaussian activated networks.

The first fact we will need is the value of the Hermite polynomials at zero, which follows from a simple computation using (1.23).

Lemma 1.14.

$$h_n(0) = \begin{cases} 0, & \text{if } n = 2k+1 \\ (-1)^k(2k-1)!!, & \text{if } n = 2k \end{cases} \tag{1.26}$$

where !! *denotes the double factorial notation.*

Lemma 1.15. $\frac{d^k}{dx^k}h_n(x) = n(n-1)\cdots(n-(k-1))h_{n-k}(x)$.

The above lemma is follows from differentiating formula (1.23).

Another well know basis for the weighted space $L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$ are given by the monomials $\{x^n\}_{n\geq 0}$. The following lemma shows how to express the Hermite polynomials in the monomials basis.

Lemma 1.16. $h_n(x) = \sum_{k=0}^{n}\binom{n}{k}h_{n-k}(0)x^k$.

The proof of the above lemma follows from the following obeservations. The Hermite polynomials are analytic functions about zero. Hence one can expand them in a Taylor series about zero, then applying lemma 1.15 gives the coefficients of the Taylor expansion leading to the formula in lemma 1.16.

Lemma 1.17.

$$\int_{\mathbb{R}} h_n(\lambda x)e^{-x^2/2}dx = \begin{cases} (2k-1)!!\sqrt{2\pi}(\lambda^{2k}+(-1)^k), & \text{if } n=2k,\ (k\geq 0) \\ 0, & \text{if } n=2k+1,\ (k\geq 0) \end{cases} \tag{1.27}$$

Proof. By using formula (1.23) it's easy to see that when n is odd, the integrand is an odd function hence the above integral is zero. For the even case, $n=2k$, we proceed as follows. Using lemma 1.16 we have

$$\int_{\mathbb{R}} h_n(\lambda x)e^{-x^2/2}dx = \sum_{i=0}^{2k}\binom{2k}{i}h_{2k-i}(0)\lambda^i\int_{\mathbb{R}} x^i e^{-x^2/2}dx.$$

We then observe that $h_{2k-i}(0)=0$ if $2k-i$ is odd and $h_{2k-i}(0)\neq 0$ is $2k-i$ is even. Write $2k-i=2j$, then the above integral becomes

$$\begin{aligned}
\sum_{i=0}^{2k}\binom{2k}{i}h_{2k-i}(0)\lambda^i\int_{\mathbb{R}} x^i e^{-x^2/2}dx &= \sum_{j=0}^{k}\binom{2k}{i}h_{2j}(0)\lambda^{2k-2j}\int_{\mathbb{R}} x^{2k-2j}e^{-x^2/2}dx \\
&= \sum_{j=0}^{k}\binom{2k}{2k-2j}(-1)^j(2j-1)!!\lambda^{2k-2j}\int_{\mathbb{R}} x^{2k-2j}e^{-x^2/2}dx \\
&= \sum_{j=0}^{k}\binom{2k}{2k-2j}(-1)^j(2j-1)!!(2k-2j-1)!!\sqrt{2\pi} \\
&= (2k-1)!!(\lambda^{2k})\sqrt{2\pi}+(-1)^k(2k-1)!!\sqrt{2\pi} \\
&= (2k-1)!!\sqrt{2\pi}(\lambda^{2k}+(-1)^k)
\end{aligned}$$

where we used lemma 1.14 to get the second inequality. □

1.5.2 Hermite Expansions

In this section we work out the Hermite expansion, see (1.24), of the functions $sin(\omega x)$ and e^{-x^2/ω^2}.

Lemma 1.18. *We have the following formulas*

$$\int_{-\infty}^{\infty}(cos(\omega x))\frac{e^{-x^2}}{\sqrt{2\pi}}dx = e^{-\omega^2/2}$$
$$\int_{-\infty}^{\infty}(sin(\omega x))\frac{e^{-x^2}}{\sqrt{2\pi}}dx = 0$$

Proof. The function $sin(\omega x))\frac{e^{-x^2}}{\sqrt{2\pi}}$ is an odd function, hence its integral over $(-\infty,\infty)$ must be zero.

To prove the first integral formula, we compute as follows

$$\begin{aligned}\int_{-\infty}^{\infty}(cos(\omega x))\frac{e^{-x^2}}{\sqrt{2\pi}}dx + i\int_{-\infty}^{\infty}(sin(\omega x))\frac{e^{-x^2}}{\sqrt{2\pi}}dx &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-x^2/2}(cos(\omega x)+isin(\omega x))dx\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-x^2/2}e^{i\omega x}dx\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-x^2/2-i\omega x}dx\\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-(x/\sqrt{2}-i\omega/\sqrt{2})^2-\omega^2/2}dx\\ &= \frac{e^{-\omega^2/2}}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-(x/\sqrt{2}-i\omega/\sqrt{2})^2}dx.\end{aligned}$$

By applying the change of variable $t = \frac{x}{\sqrt{2}} - \frac{i\omega}{\sqrt{2}}$, $dt = \frac{dx}{\sqrt{2}}$ we obtain

$$\begin{aligned}\frac{e^{-\omega^2/2}}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-(x/\sqrt{2}-i\omega/\sqrt{2})^2}dx &= \frac{e^{-\omega^2/2}}{\sqrt{2\pi}}\int_{\infty}^{\infty}e^{-t^2}\sqrt{2}dt\\ &= e^{-\omega^2/2}.\end{aligned}$$

□

Lemma 1.19. *Let $n \geq 0$ be an integer written in the form $n = 4k + l$, with $k \geq 0$ and $0 \leq l < 4$. Let a_n denote the nth Hermite coefficient of the function $\sin(\omega x)$. We have that*

$$a_n = \begin{cases} 0, & \text{if } l = 0\\ \frac{2\omega^n}{\sqrt{n!2\pi}}e^{-\omega^2/2}, & \text{if } l = 1\\ 0, & \text{if } l = 2\\ -\frac{2\omega^n}{\sqrt{n!2\pi}}e^{-\omega^2/2}, & \text{if } l = 3\end{cases} \tag{1.28}$$

Proof. Using (1.25), we see that the Hermite coefficients of $\sin(\omega x)$ are given by

$$\begin{aligned}a_n &= \int_{-\infty}^{\infty}\frac{1}{\sqrt{n!}}(-1)^n sin(\omega x)e^{-x^2/2}\frac{d^n}{dx^n}(e^{-x^2/2})\frac{e^{-x^2/2}}{\sqrt{2\pi}}dx\\ &= \frac{1}{\sqrt{n!}}(-1)^n\frac{1}{\sqrt{2\pi}}\int_{\infty}^{\infty}sin(\omega x)\frac{d^n}{dx^n}(e^{-x^2/2})dx\\ &= \frac{1}{\sqrt{n!}}(-1)^n\frac{1}{\sqrt{2\pi}}(-1)^n\int_{-\infty}^{\infty}\frac{d^n}{dx^n}(sin(\omega x))e^{-x^2/2}dx\end{aligned}$$

where the last inequality follows by integration by parts n times, using the fact that the function $sin(\omega x)\frac{d^n}{dx^n}(e^{-x^2/2})d$ decays out at $+\infty$ and $-\infty$, so the boundary components are zero.

Writing $n = 4k + l$, with $k \geq 0$ and $0 \leq l < 4$, we have that

$$\frac{d^n}{dx^n}(sin(\omega x)) = \begin{cases}\omega^n sin(\omega x), & \text{if } l = 0\\ \omega^n cos(\omega x), & \text{if } l = 1\\ -\omega^n sin(\omega x), & \text{if } l = 2\\ -\omega^n cos(\omega x), & \text{if } l = 3\end{cases} \tag{1.29}$$

By lemma 1.18, we then see that

$$a_n = \begin{cases} 0, & \text{if } l = 0 \\ \frac{2\omega^n}{\sqrt{n!2\pi}} e^{-\omega^2/2}, & \text{if } l = 1 \\ 0, & \text{if } l = 2 \\ -\frac{2\omega^n}{\sqrt{n!2\pi}} e^{-\omega^2/2}, & \text{if } l = 3 \end{cases} \tag{1.30}$$

□

We now move on to computing the Hermite expansion of a Gaussian function e^{-x^2/ω^2}.

Lemma 1.20. *The Hermite coefficients a_n of the function e^{-x^2/ω^2} are given by*

$$a_n = \begin{cases} 0, & \text{if } n = 2k+1 \\ (-1)^{2k}\left(\frac{\sqrt{2}}{\omega}\right)^{2m}\left(\frac{\omega}{\sqrt{2+\omega^2}}(2m-1)!!\sqrt{2\pi}\right)\left(\left(\frac{2}{2+\omega^2}\right)^k + (-1)^k\right), & \text{if } n = 2k. \end{cases}$$

From (1.25), we have that

$$\begin{aligned} a_n &= \int_{\mathbb{R}} e^{-x^2/\omega^2} h_n(x) e^{-x^2/2} dx \\ &= (-1)^n \int_{\mathbb{R}} e^{-x^2/\omega^2} \frac{d^n}{dx^n}(e^{-x^2/2}) dx \\ &= \int_{\mathbb{R}} \left(\frac{d^n}{dx^n}(e^{-x^2/\omega^2})\right) e^{-x^2/2} dx, \text{ on integrating by parts n times.} \end{aligned}$$

We then observe that $h_n\left(\frac{\sqrt{2}x}{\omega}\right) = (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n e^{x^2/\omega^2} \frac{d^n}{dx^n}(e^{-x^2/\omega^2})$. Substituting this into the above we obtain

$$\begin{aligned} a_n &= (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n \int_{\mathbb{R}} h_n\left(\frac{\sqrt{2}x}{\omega}\right) e^{-x^2/\omega^2} e^{-x^2/2} dx \\ &= (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n \int_{\mathbb{R}} h_n\left(\frac{\sqrt{2}x}{\omega}\right) e^{-\left(\frac{2+\omega^2}{2\omega^2}\right)x^2} dx. \end{aligned}$$

Using the substitution $y = \left(\frac{2+\omega^2}{\omega^2}\right)^{1/2} x$, with $dy = \left(\frac{2+\omega^2}{\omega^2}\right)^{1/2} dx$, we have that

$$\begin{aligned} (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n \int_{\mathbb{R}} h_n\left(\frac{\sqrt{2}x}{\omega}\right) e^{-\left(\frac{2+\omega^2}{2\omega^2}\right)x^2} dx &= (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n \int_{\mathbb{R}} h_n\left(\frac{\sqrt{2}\omega}{\omega\sqrt{2+\omega^2}} y\right) e^{-y^2/2} \frac{\omega}{\sqrt{2+\omega^2}} dy \\ &= (-1)^n \frac{1}{\sqrt{n!}}\left(\frac{\omega}{\sqrt{2}}\right)^n \frac{\omega}{\sqrt{2+\omega^2}} \int_{\mathbb{R}} h_n\left(\frac{\sqrt{2}}{\sqrt{2+\omega^2}} y\right) dy. \end{aligned}$$

This latter integral can be computed using lemma 1.17. We thus obtain

$$a_n = \begin{cases} 0, & \text{if } n = 2k+1 \\ (-1)^{2k}\left(\frac{\sqrt{2}}{\omega}\right)^{2m}\left(\frac{\omega}{\sqrt{2+\omega^2}}(2m-1)!!\sqrt{2\pi}\right)\left(\left(\frac{2}{2+\omega^2}\right)^k + (-1)^k\right), & \text{if } n = 2k. \end{cases}$$

We will need the following lemma, see lemma D.3 in [13].

Lemma 1.21. *Let $X = [x_1, \ldots, x_N]^T$ in$\mathbb{R}^{N\times d}$ where $||x_i||_2 = \sqrt{d}$ for $1 \leq i \leq N$. Let*

$$G = \mathbb{E}_{w\sim\mathcal{N}(0,\frac{1}{d}\mathbb{I}_d)}\left(\phi(Xw)\phi(Xw)^T\right),$$

where $\phi(x)$ is either $sin(\omega x)$ or e^{-x^2/ω^2}. Let a_n denote the nth coefficient of the Hermite expansion of ϕ. Then

$$G = \sum_{n=0}^{\infty} \frac{a_n^2}{d^n}\left(XX^T\right)^{\circ_n} \tag{1.31}$$

where $\circ_n$ denotes the n-fold Hadamard product, and the "=" is to be interpreted as uniform convergence.

The proof of the following lemma follows from lemma 3.4 in [13] on noting that $sin(\omega x), e^{-x^2/\omega^2} \in L^2(\mathbb{R}, \frac{e^{-x^2/2}}{\sqrt{2\pi}})$.

Lemma 1.22. *Let $X \in \mathbb{R}^{N\times d}$ be random sub-Gaussian matrix whose rows are i.i.d sub-Gaussian vectors with $||X_{i:}||_2 = \sqrt{n_0}$ and $||X_{i:}||_{\psi_2} \leq c$ for all $1 \leq i \leq N$, where $c > 0$ is some constant independent of n_0. Fix an integer $k \geq 2$, then if $N \leq d^k$, we have $\mathbb{P}(\sigma_{min}(X^{*k}) \geq \frac{d^{k/2}}{2}) \geq 1 - 2N^2e^{-c_1dN^{-2/k}}$, where $c_1 > 0$, where $\phi(x) = sin(\omega x)$ or e^{-x^2/ω^2}.*

We now prove the following theorem that computes a lower bound on $\lambda_{\min}(G)$, where $G = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\right)$. This theorem was originally proved for non-linearities satisfying $|\phi(x)| \leq |x|$ in [13], see theorem 3.3. Our proof will follow their proof using lemmas 1.19, 1.20.

Theorem 1.23. *Let X satisfy the same data assumptions as in lemma 1.22. Let ϕ denote one of the activations $sin(\omega x)$ or e^{-x^2/ω^2} with a_n denoting the Hermite coefficients in a Hermite expansion of ϕ. Let $G = \mathbb{E}_{w\sim\mathcal{N}(0,\beta_1^2)}\left(\phi(Xw)\phi(Xw)^T\right)$. Fix an integer $n \geq 2$. Then for $N \leq d^k$,*

$$\mathbb{P}\left(\lambda_{\min}(G) \geq \frac{a_n^2}{8}\right) \geq 1 - 2N^2e^{-c_1N^4/5k}\Big).$$

In other words $\lambda_{\min}(G) = \Omega(1)$ w.p. $\geq 1 - 2N^2e^{-c_1N^4/5k}$.

Proof. By lemmas 1.19, 1.20 we have that the coefficient a_n of the Hermite expansion of $sin(\omega x)$ is non-zero for $n \equiv 1$ or $3 mod(4)$ and is non-zero for $n \equiv 0$ or $mod(2)$ for e^{-x^2/ω^2}. In particular, in both cases we can find $n \geq 10k$ such that $a_n \neq 0$. Applying lemma 1.22, we see that for $N \leq d^n$

$$\lambda_{\min}((XX^T)^{\circ_n}) = \sigma^2_{\min}(X^{*n}) \geq \frac{d^n}{4} \tag{1.32}$$

w.p. $\geq 1 - 2N^2e^{-c_1dN^{-2/n}} \geq 1 - 2N^2e^{-c_1N^4/5k}$, using the fact that $N \leq d^n$ and $n \geq 10k$.
Using lemma 1.21 we see that we can write

$$G = \sum_{n=0}^{\infty} \frac{a_n^2}{d^n}\left(XX^T\right)^{\circ_n}. \tag{1.33}$$

If we let $S_m = \sum_{i=0}^{m} \frac{a_i^2}{d^i}(XX^T)^{\circ_i}$ denote the mth partial sum then we see by (1.33) that there exists $m \geq n$ such that

$$||G - S_m||_F \leq \frac{a_n^2}{2d^n}\lambda_{\min}((XX^T)^{\circ_n}).$$

As $\lambda_{\min}(S_m) \geq \lambda_{\min}(S_n) \geq \frac{a_n^2}{d^n}\lambda_{\min}((XX^T)^{\circ_n})$. Thus on an application of Weyl's inequality, we get

$$\lambda_{\min}(G) \geq \lambda_{\min}(S_m) - \left(\frac{a_n^2}{2d^n}\lambda_{\min}((XX^T)^{\circ_n})\right) \geq \frac{a_n^2}{2d^n}\lambda_{\min}((XX^T)^{\circ_n} \geq \frac{a_n^2}{8} > 0$$

w.p. $\geq 1 - 2N^2e^{-c_1N^4/5k}$, where the third inequality follows by applying (1.32). □

1.6. Proofs of theorems in sec. 4 of the main paper

1.6.1 Proofs of thms. 4.2 and 4.4 from the main paper

In this section we want to give the proofs of **thms. 4.2, 4.4** from the main paper. The proofs of **thms. 4.7, 4.9** from the main paper will easily follow from the proofs of these results.

We will start by giving the proof of the **thm. 4.2** from the main paper which is a result about shallow networks i.e. networks with 1 hidden layer.

We will be making the following standard assumptions. (i) the data samples X will all be sub-gaussian random vectors with $||X_i||_2 \leq 1$ and $||X_i||_{\psi_2} = \mathcal{O}(1)$, for the definition of the norm $||\cdot||_{\psi_2}$ please see sec. 1.4, (ii) the labels Y satisfy $||Y_i||_2 = \mathcal{O}(1)$ for all $i \in [N]$. Furthermore, the final output width n_2 will be fixed and not varied

The approach taken to prove the main theorems of **sec. 4** of the main paper is to show that the conditions (1.6)-(1.8) are satisfied.

Proof of thm. 4.2 from the main paper. Taking $C_1 = C_2 = 1$, equations (1.6)-(1.8) are

$$\sigma_0^2 \geq 16(\max\{\sqrt{An_0N}, \sqrt{An_1N}\}\overline{\lambda}_2\sqrt{2\mathcal{L}(\theta_0)} \tag{1.34}$$

$$\sigma_0^3 \geq 32\overline{B}^2||X||_F\sqrt{An_0N}\overline{\lambda}_1\overline{\lambda}_2\sqrt{2\mathcal{L}(\theta_0)} \tag{1.35}$$

$$\sigma_0^2 \geq 8\overline{B}^4||X||_F\sqrt{An_0N}\overline{\lambda}_1\overline{\lambda}_2 \tag{1.36}$$

Recall that $W_1^0 \in \mathbb{R}^{n_0\times n_1}$ and $W_2^0 \in \mathbb{R}^{n_1\times n_2}$. Since our last layer width n_2 is fixed, Theorem 2.13 of [4] implies w.p $\geq 1 - e^{-\Omega(n_1)}$

$$\overline{\lambda}_1 = \mathcal{O}(1), \overline{\lambda}_2 = \mathcal{O}\left(\frac{\sqrt{n_2}}{\sqrt{n_1}}\right). \tag{1.37}$$

By proposition 1.9, given any $\delta > 0$ we have that w.p $\geq 1 - \delta$ that

$$\sigma_0 \geq \left(\frac{n_1\lambda}{4}\right)^{1/2} \tag{1.38}$$

if $n_1 \geq \widetilde{\Omega}(\frac{N}{\lambda})$, where

$$\lambda = \lambda_{min}\left(\mathbb{E}_{w\sim\mathcal{N}(0,(\omega\sqrt{n_1})^{-1}I_{\sqrt{n_0}})}[\phi(wX)\phi(wX)^T]\right).$$

Furthermore, we have that

$$\sqrt{2\mathcal{L}(\theta_0)} = \mathcal{O}(\sqrt{N})$$

which follows from lemma 1.11. Finally, using lemma 1.23 we have that $\lambda \geq \Omega(1)$. Thus we find that equations (1.34)-(1.36) are satisfied if $n_1 = \Omega(N^{3/2})$. It follows by thm. 1.6 that gradient descent converges to a global minimum for a small enough learning rate, and the proof is complete. □

Remark 1.24. The statement of the **thm. 4.2** in the main paper was stated for the activations Gaussian, sine and sinc and not wavelet. The reason for this was that the wavelet activation was a complex valued activate give by $e^{i\omega_0 x}e^{-x^2/\omega_1^2}$, where i denotes the complex imaginary number. However, note that the absolute value of this activation is given by the Gaussian term: e^{-x^2/ω_1^2} and hence it is easy to see that it satisfies (1.4). Therefore, the above proof goes through with no issues for this activation as well.

Remark 1.25. The proof was given for the LeCun's normal initialization however the same proof also goes through for the Kaiming normal [7] and Xavier normal [6] initializations.

We now move on to proving **thm. 4.4** from the main paper. In order to do this we will impose the same conditions we did in the above proof of **thm. 4.2** from the main paper. Furthermore, we will also assume $\sigma_0(F_{k-1})^2 = \Omega(n_{L-1})$ w.h.p. This has been proven for ReLU networks, see theorem 5.1 in [12], and for shallow Gaussian, sine, sinc, wavelet activated networks, see lems. 1.23 in sec. 1.5.2

Proof of thm. 4.4 from the main paper. The goal is to show that inequalities (1.6)-(1.8) hold.

Using the notation of theorem 1.6, we set $C_l = 1$ for all $l \in [L]$. Applying theorem 2.13 from [4] and theorem 4.4.5 of [22], we find that

$$\overline{\lambda}_L = \mathcal{O}(\frac{\sqrt{n_L}}{\sqrt{n_{L-1}}}) \tag{1.39}$$

$$\overline{\lambda}_{L-1} = \mathcal{O}(\frac{\sqrt{n_{L-1}} + \sqrt{n_{L-2}}}{\sqrt{n_{L-2}}}) \tag{1.40}$$

$$\overline{\lambda}_l = \mathcal{O}(1), \text{ for } l \in [2, L-2] \tag{1.41}$$

$$\overline{\lambda}_1 = \mathcal{O}(\frac{\max\{\sqrt{m}, \sqrt{n_0}\}}{\sqrt{n_0}}). \tag{1.42}$$

We check inequality (1.6). We need to show that the following inequality holds

$$\sigma_0^2 \geq 16\overline{B}^{L-2}\sqrt{An_{L-1}N}(\overline{\lambda}_{L-1})(\overline{\lambda}_L)\sqrt{2\mathcal{L}(\theta_0)}.$$

Applying the asymptotics (1.39), (1.40) and the fact that $\sqrt{2\mathcal{L}(\theta_0)} = \mathcal{O}(\sqrt{N})$, see lemma 1.11, we see that the left hand side of the above inequality has order $\mathcal{O}(N\sqrt{n_{L-1}})$. From our assumption on σ_0, and the assumptions on the width of the network in the statement of **thm. 4.4** from the main paper, we see that the above inequality holds. We move on to checking inequality (1.7). We need to show that

$$\sigma_0^3 \geq 32\sqrt{2\mathcal{L}(\theta_0)}\Bigg(\overline{B}^{2L-2}||X||_F(\sqrt{An_0N})\overline{\lambda}_{2\to L}\prod_{j=1}^{L-1}\overline{\lambda}_j + \sum_{i=2}^{L-1}\overline{B}^{2(L-i)-1}(An_{i-1}N)\overline{\lambda}_{i+1\to L}\prod_{j=i+1}^{L-1}\overline{\lambda}_j\Bigg).. \tag{1.43}$$

Using the asymptotics (1.39)-(1.42) we see that the left hand side of the above inequality has order $\mathcal{O}(N\sqrt{n_{L-1}})$. Using our assumption that $n_{L-1} = \Omega(N^{5/2})$ and the assumption on σ_0, we see that that the inequality 1.43 is satisfied.

A similar analysis shows that (1.8) is satisfied, and the proof of the theorem is complete. □

As in rmk. 1.24 the above proof can be easily made to work for the wavelet activation function.

Remark 1.26. The proof was given for the LeCun's normal initialization however the same proof also goes through for the Kaiming normal [7] and Xavier normal [6] initializations.

1.6.2 Proofs of thms. 4.7 and 4.9 from the main paper

We observe that in the previous section, the proof of **thm. 4.2** and **thm. 4.4** in the main paper involved using the random matrix theory of matrices with entries from a normal distribution.

In the shallow case we by using theorem 2.13 of [4] we were able to obtain the complexity equality

$$\lambda_L = \mathcal{O}(\frac{\sqrt{n_L}}{\sqrt{n_{L-1}}}) \tag{1.44}$$

when we initialized the final layer with weights randomly chosen for a normal distribution of the form $\mathcal{N}(0, 1/n_{L-1})$. We now observe that if we initialize the final layer weights using the distribution $\mathcal{N}(0, 1/n_{L-1}^p)$ for $p \geq 1$. Then applying theorem 2.13 of [4] gives

$$\lambda_L = \mathcal{O}\Bigg(\frac{\sqrt{n_L}}{n_{L-1}^{p/2}}\Bigg). \tag{1.45}$$

If we then go through the proof of **thm. 4.2** of the main paper given in the previous section we see that the conditions (1.6)-(1.8) are easier to satisfy. This is the premise of the proofs of **thm. 4.7** and **thm. 4.9** from the paper.

Proof of thm. 4.7 from the main paper. Taking $C_1 = C_2 = 1$, equations (1.6)-(1.8) are

$$\sigma_0^2 \geq 16(\max\{\sqrt{An_0N}, \sqrt{An_1N}\}\overline{\lambda}_2\sqrt{2\mathcal{L}(\theta_0)} \tag{1.46}$$

$$\sigma_0^3 \geq 32\overline{B}^2||X||_F\sqrt{An_0N}\overline{\lambda}_1\overline{\lambda}_2\sqrt{2\mathcal{L}(\theta_0)} \tag{1.47}$$

$$\sigma_0^2 \geq 8\overline{B}^4||X||_F\sqrt{An_0N}\overline{\lambda}_1\overline{\lambda}_2. \tag{1.48}$$

Applying theorem 2.13 of [4] we obtain w.p $\geq 1 - e^{-\Omega(n_1)}$

$$\overline{\lambda}_1 = \mathcal{O}(1), \overline{\lambda}_2 = \mathcal{O}\bigg(\frac{\sqrt{n_2}}{{n_1}^{3/4}}\bigg). \tag{1.49}$$

By proposition 1.9, given any $\delta > 0$ we have that w.p $\geq 1 - \delta$ that

$$\sigma_0 \geq \bigg(\frac{n_1\lambda}{4}\bigg)^{1/2} \tag{1.50}$$

if $n_1 \geq \Omega(\frac{N}{\lambda})$, where

$$\lambda = \lambda_{min}\big(\mathbb{E}_{w\sim\mathcal{N}(0,(\omega\sqrt{n_1})^{-1}I_{\sqrt{n_0}})}[\phi(wX)\phi(wX)^T]\big).$$

Furthermore, we have that

$$\sqrt{2\mathcal{L}(\theta_0)} = \mathcal{O}\big(\sqrt{N}\big)$$

which follows from lemma 1.11. Finally, using lemma 1.23 we have that $\lambda \geq \Omega(1)$. Thus we find that equations (1.34)-(1.36) are satisfied if $n_1 = \Omega(N)$. It follows by thm. 1.6 that gradient descent converges to a global minimum for a small enough learning rate, and the proof is complete.

□

Proof of thm. 4.9 from the main paper. The proof follows the approach taken in the above proof. The starting point is to note that the goal is to show that inequalities (1.6)-(1.8) hold.

Using the notation of theorem 1.6, we set $C_l = 1$ for all $l \in [L]$. Applying theorem 2.13 from [4] and theorem 4.4.5 of [22], we find that

$$\overline{\lambda}_L = \mathcal{O}(\frac{\sqrt{n_L}}{n_{L-1}^{3/4}}) \tag{1.51}$$

$$\overline{\lambda}_{L-1} = \mathcal{O}(\frac{\sqrt{n_{L-1}} + \sqrt{n_{L-2}}}{\sqrt{n_{L-2}}}) \tag{1.52}$$

$$\overline{\lambda}_l = \mathcal{O}(1), \text{for } l \in [2, L-2] \tag{1.53}$$

$$\overline{\lambda}_1 = \mathcal{O}(\frac{\max\{\sqrt{m}, \sqrt{n_0}\}}{\sqrt{n_0}}). \tag{1.54}$$

The proof then follows in exactly the same way as in the proof of **thm. 4.4** from the main paper shown in sec. 1.6.2. □

As in rmk. 1.24 the above proof can be easily made to work for the wavelet activation function.

1.7. Learning rate

The statements of **thms. 4.2., 4.4., 4.7, 4.9** from the main paper included the condition that the learning rate had to be small enough. A natural question that arises is how small the learning rate has to be and whether we are able to obtain any useful bounds for the learning rate.

We observed from secs. 1.6.1 and 1.6.2 that the main ingredient to the proofs of **thms. 4.2., 4.4., 4.7, 4.9** from the main paper was to establish the inequalities (1.6)-(1.8) from thm. 1.6. Going back to thm. 1.6 we see that the learning rate must satisfy the following inequality:

$$\eta < \min\bigg\{\frac{4}{\sigma_0^2}, (\overline{\lambda}_L)^{-3}\bigg(B^{2L}||X||_F\sqrt{An_oN}(\overline{\lambda}_1)(\overline{\lambda}_{2\to L-1})^2 + \sum_{i=2}^{L} B^{2L}(An_{i-1}N)(\overline{\lambda}_{i+1\to L-1})^2\bigg)^{-1}\bigg\}. \tag{1.55}$$

While it is hard to work with this bound due to the second term in the brackets we do see that that the learning rate depends on the various quantities associated to the training of the network, such as the norm of the data set X, the operators norm of the initial weight matrices and the minimum singular value of the output of the final hidden layer. Using (1.55) we can obtain the bound

$$\eta \leq \frac{4}{\sigma_0^2}. \tag{1.56}$$

If we then apply proposition 1.9, given any $\delta > 0$ we have that w.p $\geq 1 - \delta$ that

$$\sigma_0 \geq \left(\frac{n_1 \lambda}{4}\right)^{1/2}. \tag{1.57}$$

Substituting this back into (1.56) we obtain

$$\eta \leq \frac{C}{n_1}, \tag{1.58}$$

where $C > 0$ is a constant that is independent of the widths of the network. We thus see that the learning rate must be bounded above by the inverse of the width. This implies that for very large widths, the learning rate will have to be extremely small. However, in practise we find that one can take much bigger learning rates. This is a limitation of the theory in that it requires very small learning rates. This is however common amongst theoretical works establishing convergence for gradient descent.

1.8. Extending the theory to Adam

So far all the theory we have talked about, including the statements of **thms. 4.2, 4.4, 4.7 and 4.9** from the main paper, has been for gradient descent. However, in the area of Neural Fields the more common optimizer that is used is the Adam optimizer. In this section, we will briefly show how to extend the theory to the case of the Adam optimizer. The reader who is not familiar with the Adam optimizer is kindly asked to consult the original reference [8].

The Adam optimizer involves two main terms. The first one is a first moment estimate given by

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{1.59}$$

where g_t denotes the gradient at iteration t and β_1 is a moving average parameter that is usually taken to be 0.9. The second one is a second moment estimate given by:

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{1.60}$$

where β_2 is another parameter that is usually fixed at 0.999.

These moment estimates then undergo a bias corrected normalization leading to

$$\hat{m_t} = \frac{m_t}{(1 - \beta_1)^t} \text{ and } \hat{v_t} = \frac{v_t}{(1 - \beta_2)^t}. \tag{1.61}$$

Finally, if we denote the weights at iteration t by W_t. Then the Adam optimizer update is given by

$$W_t = W_{t-1} - \frac{\eta}{\sqrt{\hat{v_t} + \epsilon}} \hat{m_t} \tag{1.62}$$

where $\epsilon > 0$ is a stability factor so that the algorithm doesn't encounter a division by zero.

We now observe that as is we take more and more iterates of the algorithm the quantity $\frac{1}{||\sqrt{\hat{v_t}+\epsilon}||}$ must be bounded by some uniform constant $C > 0$. This is because as $t \to \infty$ if $v_t \to 0$ then $\frac{1}{||\sqrt{\hat{v_t}+\epsilon}||}$ will always be bounded above $\frac{2}{\epsilon}$. If on the other $v_t \to \infty$ then $\frac{1}{||\sqrt{\hat{v_t}+\epsilon}||}$ is bounded above by 1. We thus see that we always have that the quantity $\frac{1}{||\sqrt{\hat{v_t}+\epsilon}||}$ must be bounded during all the steps of the algorithm by a uniform constant.

We now go back to the proof of thm. 1.6. The main part of this proof was to establish a bound on the difference $\sum_{s=0}^{k} ||W_l^{s+1} - W_l^s||_F$. This was done by using the gradient descent update step and then appealing to lem. 1.4. The

difference now is that we have to use the Adam update (1.62). From what we said about the term $\frac{1}{||\sqrt{\hat{v}_t}+\epsilon||}$ always being bounded by a constant $c > 0$. We can then estimate the sum $\sum_{s=0}^{k} ||W_l^{s+1} - W_l^s||_F$ as follows:

$$\sum_{s=0}^{k} ||W_l^{s+1} - W_l^s||_F \leq C\eta \sum_{s=0}^{k} ||\hat{m}_t||_F \tag{1.63}$$

$$\leq C\widetilde{C}_1(\beta_1)\eta \sum_{s=0}^{k} ||m_t||_F \tag{1.64}$$

$$\leq C\widetilde{C}_1(\beta_1)\widetilde{C}_2(\beta_2) \sum_{s=0}^{k} ||\nabla_{W_l}\mathcal{L}||_F \tag{1.65}$$

where $\widetilde{C}_1(\beta_1)$ and $\widetilde{C}_2(\beta_1)$ are constants that depend on β_1.

We thus see that the proof of thm. 1.6 goes through though now the inequalities (1.6)-(1.8) will involve constants on the right handside that depend on β_1. However, the proofs of **thms. 4.2, 4.4, 4.7, 4.9** from the main paper, given in secs. 1.6.1 and 1.6.2, were all based on complexity estimates. Thus an extra constant depending on β_1 on the right hand side of inequalities (1.6)-(1.8) will not affect the complexity bounds that were used and hence **thms. 4.2, 4.4, 4.7, 4.9** go through for the Adam optimizer when training with full batch.

2. Experiments: Applications of Neural Fields to Vision

2.1. Hardware and Software

All experiments were run on a Nvidia RTX A6000 GPU. Furthermore, all the experiments were coded in PyTorch version 2.0.1.

Hyperparameters: Each of the activation functions Gaussian, sine, sinc and wavelet all had an extra hyperparameter which consisted of either a frequency component or a variance component or as in the case of a wavelet both. So as to obtain the best results for our experiments and to compare with results in the literature we ran sweeps for each of these parameters and have picked the best ones. Our values conincide with those from the literature. We list these parameters for the sake of completeness:

1. For Gaussian activations we found the best variance to be 0.1^2. This fits with what was found in [2, 15, 18].
2. For a sinc activation we found the best frequency to be 8, which fits with [16].
3. For a sine activation we found that a frequency between $20 - 30$ performed best with in general 30 leading to slightly higher PSNR values for images. This also fits with [19] where it is suggested that anywhere between $10 - 30$ should work well.
4. For the wavelet activation we found the best frequency to be 10 and the variance to be 0.05^2 which fits what was found in [17].

Optimizers: We will only use two optimizers throughout all experiments. Namely, Gradient Descent (GD) and Adam. We did a sweep for the learning rate and found 1e-2 for GD to work best and 1e-4 to work best for Adam. Note that we kept these learning rates fixed through all experiments so as to yield fair results that could be compared.

2.2. Initializations we will be using

In total we will be experimenting with various initializations in the experiments. We will first outline the normal initializations we will be using:

Normal Initializations:

$$\textbf{LeCun Normal Initialization [9]: } (W_l^0)_{ij} \sim \mathcal{N}(0, 1/n_{l-1}), \text{ for } l \in [L]. \tag{2.1}$$

$$\textbf{Kaiming Normal Initialization [7]: } (W_l^0)_{ij} \sim \mathcal{N}(0, 2/n_{l-1}), \text{ for } l \in [L]. \tag{2.2}$$

$$\textbf{Xavier Normal Initialization [7]: } (W_l^0)_{ij} \sim \mathcal{N}(0, 2/(n_l + n_{l-1})), \text{ for } l \in [L]. \tag{2.3}$$

$$\textbf{Initialization 1 (ours): } (W_l^0)_{ij} \sim \mathcal{N}(0, 1/n_{l-1}) \text{ for } l \in [L-1]. \text{ and } (W_L^0)_{ij} \sim \mathcal{N}(0, 2/(n_{l-1}^{3/2})). \tag{2.4}$$

For details on the motivation of initialization 1 please see **sec. 4.4** of the main paper.

For the image regression experiments we will also compare with the following initializations as the founders of these initializations proved gradient descent converges to a global minimum when initialized with on of these.

$$\textbf{Du et al. Initialization [5]: } (W_l^0)_{ij} \sim \mathcal{N}(0, 1), \text{ for } l \in [L]. \tag{2.5}$$

$$\textbf{Arora et al. Initialization [1]: } (W_l^0)_{ij} \sim \mathcal{N}(0, 1/4), \text{ for } l \in [L]. \tag{2.6}$$

Arora et al. [1] proved their gradient convergence theorem for any for initializations of the form $\mathcal{N}(0, \kappa^2)$ where $0 < \kappa < 1$. We decided to simply pick a value of κ that trained reasonably and thus went with $1/4$.

For all the above normal initializations the biases are initialized to zero.

Remark 2.1. In our initialization 1, we note that there is a 2 in the numerator of the standard deviation of the final layer normal distribution we are sampling the final layer weights from. The reader may be wondering why the factor of 2? The theory developed so far is really a complexity theory, detailing how quantities should scale. This means the theory in general can only predict outcomes upto a constant. We thus introduced the 2 as we found it slightly trained better than if we were to have a 1 there. However, even if we were to use a 1 in the numerator we still found it did much better than existing initializations within the literature.

Uniform Initializations: We will also be experimenting with uniform initializations:

$$\textbf{LeCun Uniform: } (W_l^0)_{ij} \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}), \text{ for } l \in [L]. \text{ Biases are initialized to zero .} \tag{2.7}$$

$$\textbf{Kaiming Uniform: } (W_l^0)_{ij} \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}) \tag{2.8}$$

$$b_l^0 \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}), \text{ for } l \in [L]. \tag{2.9}$$

$$\textbf{Xavier Uniform: } (W_l^0)_{ij} \sim \mathcal{U}(-\sqrt{6}/\sqrt{n_{l-1} + n_l}, \sqrt{6}/\sqrt{n_{l-1} + n_l}) \tag{2.10}$$

$$b_l^0 \sim \mathcal{U}(-1/\sqrt{n_{l-1} + n_l}, 1/\sqrt{n_{l-1} + n_l}), \text{ for } l \in [L]. \tag{2.11}$$

$$\textbf{Initialization 2 (ours): } (W_l^0)_{ij} \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}) \text{ for } l \neq L. \tag{2.12}$$

$$(W_L^0)_{ij} \sim \mathcal{U}(-1/(n_{l-1}^{3/4}), 1/(n_{l-1}^{3/4})). \tag{2.13}$$

For our above initialization 2 we found that initializing the biases to zero worked well. Otherwise the following initialization for the biases also worked well:

$$\textbf{Initialization 2 for biases: } b_l^0 \sim \mathcal{U}(-1/\sqrt{n_{l-1}}, 1/\sqrt{n_{l-1}}) \text{ for } l \neq L. \tag{2.14}$$

$$b_L^0 \sim \mathcal{U}(-1/(n_{l-1}^{3/4}), 1/(n_{l-1}^{3/4})). \tag{2.15}$$

Finally we will also need to compare our initialization for a sine activation with the principled scheme obtained in Sitzmann et al. [19] for SIRENs. One of the key ideas we want to show here is how effective our initialization is in that it can also be used as a simple way to adjust an already practical initialization scheme. Thus we retain everything the original SIREN initialization scheme does except that we simply sample from a smaller variance uniform distribution on the final layer.

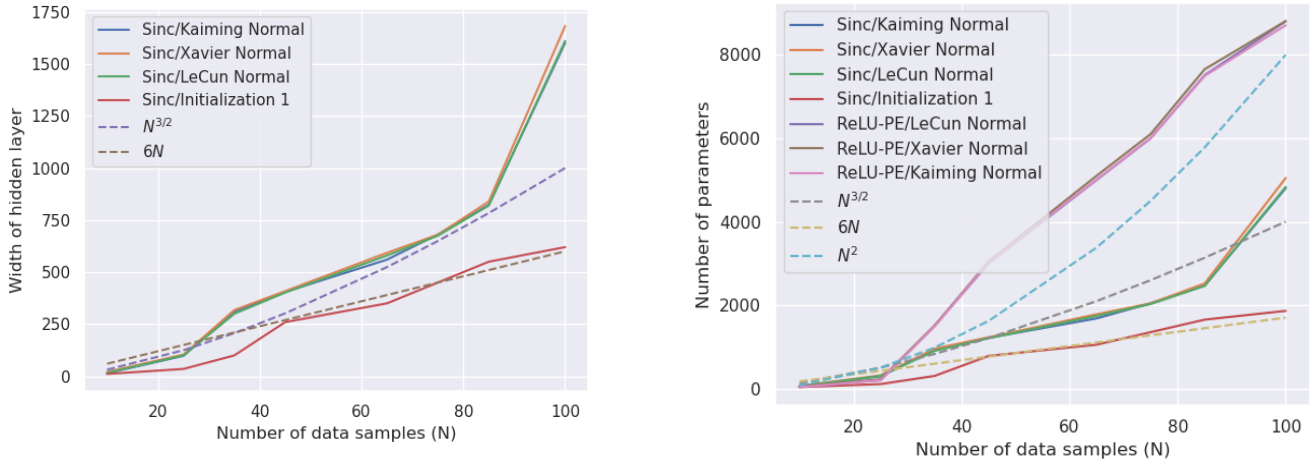

Figure 1. **Left;** Results from initializing four sinc activated networks with different initializations and trained with gradient descent on various data sample sizes till the networks reach a PSNR of 50dB. As the number of samples go up the networks initialized with LeCun, Kaiming, Xavier Normal initializations all need much more width scaling faster than $N^{3/2}$. While the one initialized with initialization 1 needs much less, scaling like $6N$. **Right;** The same experiment as the left though now we measure total parameters of the network and look at ReLU-PE networks. In this case we see that the ReLU-PE network scaled faster than N^2. We thus see that in both experiments initialization 1 with a sinc activated network is far more parameter efficient.

Initialization for SIREN's:

$$\textbf{Initialization 3 (ours):}\ (W_l^0)_{ij} \sim \mathcal{U}(-\sqrt{6}/(\sqrt{n_{l-1}}\omega), \sqrt{6}/(\sqrt{n_{l-1}}\omega)) \text{ for } l \neq L. \tag{2.16}$$

$$(W_L^0)_{ij} \sim \mathcal{U}(-\sqrt{6}/(n_{l-1}^{3/4}\omega), \sqrt{6}/(n_{l-1}^{3/4}\omega)) \tag{2.17}$$

where ω denotes the frequency of the sine activation. For this initialization all biases will be initialized to zero.

We will be comparing this initialization 3 to the initialization in [19] which we shall simply call the SIREN intialization.

2.3. Testing The Theory

Shallow networks: We performed a 1-dimensional curve fitting experiment on the function $f(x) = \sin(2\pi x) + \sin(6\pi x) + \sin(10\pi x)$. We systematically sampled the curve at intervals of 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 points, creating 10 datasets of varying sizes for our training data.

In the first experiment we wanted to compare how much width is needed in the hidden layer for the network to achieve a PSNR of 50dB when trained with gradient descent on each of the datasets above. We compared four networks each admitting a sinc activation with frequency 8 such that each one was initialized with Kaiming Normal, LeCun Normal, Xavier Normal and Initialization 1 (see sec. 2.2). **Thm. 4.2** from the main paper predicts that as the number of samples increase the width of the network should scale super-linearly when initializaed with Kaiming Normal, LeCun Normal, Xavier Normal and **thm. 4.7** predicts the network should scale linearly when initialized with initialization 1. To test this, we plotted two curves, namely the function $y = N^{3/2}$ and the function $y = 6N$. Fig. 1 (left) shows the result of the experiments. We observe that the sinc network initialized with our initialization 1 needs much less width and roughly scales according to $6N$. Furthmore, we see that the other 3 initialized networks need much more width and scale greater than $N^{3/2}$. This verifies that there is merit in **thms. 4.2 and 4.7** from the main paper. Observe that we did not include ReLU-PE in this experiment as the input dimension of a ReLU-PE layer would change due to the positional embedding layer, therefore the comparison wouldn't be fair.

The experiment was repeated though now we counted the total number of parameters needed for the network to converge to a PSNR of 50dB. In this case, we could include a ReLU-PE network trained on various initializations. As can be seen by fig. 1 (right) initialization 1 scales the slowest with respect to data size, making it much more efficient in term of parameters for gradient descent to converge to a high PSNR value.

Figure 2. A $181 \times 213 \times 3$ peppers image used for the image experiments in sec. 2.3 and 2.4.

Deep networks: For the case of deep networks we ran a similar experiment to the above shallow networks except that this time we used an image regression task. In this case we took a moderately difficult image, namely a Peppers Image, see fig 2, which is a $181 \times 213 \times 3$ image. We then sampled $100, 200, 300, 400, 500, 600, 700, 800, 900, 1000$ points via a uniform distribution creating a total of 10 datasets of varying size. Similar to the shallow experiment above we then considered four sinc deep network, each with 2 hidden layers, and initialized with LeCun, Kaiming, Xavier Normal and initialization 1. We trained these networks till they converged to a PSNR of 50dB under gradient descent only allowing an increase in the final hidden layer width. **Thm. 4.4** from the main paper predicts that the networks employing the LeCun, Kaiming, Xavier Normal initializations should all scale super-quadratically with the number of samples, and **thm. 4.9** from the main paper predicts the one initialized with initialization 1 should scale quadratically. We found that the functions $y = (1/4000)N^{5/2}$ was a good predictor for the amount of width needed by the sinc networks initialized by LeCun, Kaiming, Xavier Normal and the function $y = (1/290)N^2$ was a good predictor for the amount of width needed for the sinc network initialized with initialization 1. Fig. 3 (left) shows the results of this experiment.

We then tested the experiment on total number of parameters and included ReLU-PE into the mix. In this case the ReLu-PE scaled cubically with respect to the data size, showing that it is an extremely parameter inefficient activation. Fig. 3 (right) shows the results of that experiment.

The above experiment was repeated with a four hidden layer deep network. In this setting we sampled $200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000$ points via Gaussian distribution centred at the centre of the image. In this case too we found that **thm. 4.4** and **thm. 4.9** predictions were accurate. Fig. 4 shows the results of this experiment.

We thus see from these experiments that the sinc networks that were initialized with initialization 1, see sec. 2.2, were much more parameter efficient for gradient descent to converge to a high PSNR.

2.4. Image Reconstruction

Image reconstruction is the following problem: Given pixel coordinates $\mathbf{x} \in \mathbb{R}^2$, we aim to optimize the network f to regress the associated RGB values $\mathbf{c} \in \mathbb{R}^3$.

We will consider two images for this take both sampled fully. We will use the peppers image 2 as well as a slightly more complicated Lion image as shown in fig. 5.

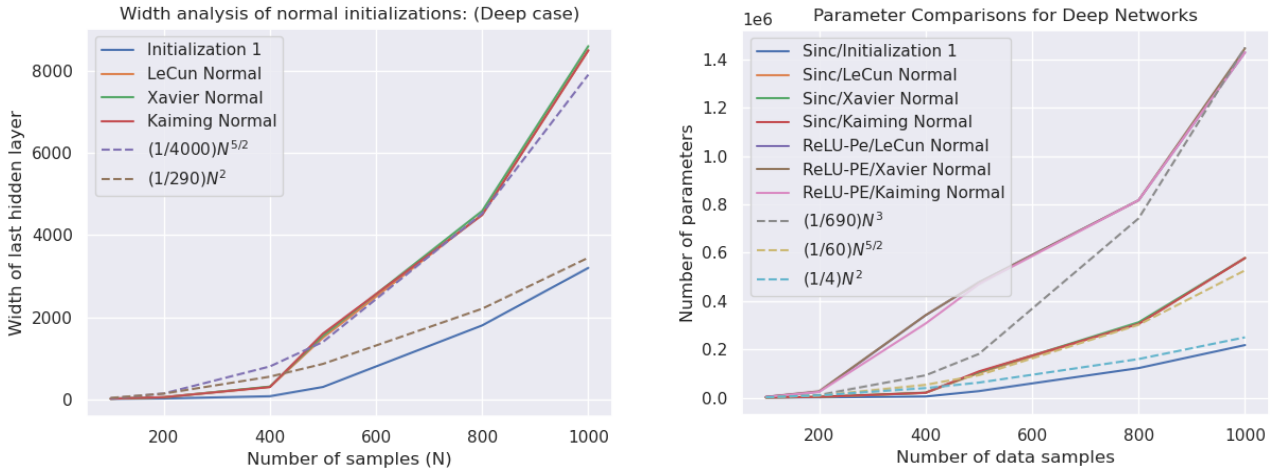

Figure 3. **Left;** Results from initializing four sinc activated networks with different initializations and trained with gradient descent on various image data sample sizes till the networks reach a PSNR of 50dB. As the number of samples go up the networks initialized with LeCun, Kaiming, Xavier Normal initializations all need much more width scaling faster than $(1/4000)N^{5/2}$. While the one initialized with initialization 1 needs much less, scaling like $(1/290)N^2$. **Right;** The same experiment as the left though now we measure total parameters of the network and look at ReLU-PE networks. In this case we see that the ReLU-PE network scaled faster than $(1/690)N^3$. We thus see that in both experiments initialization 1 with a sinc activated network is far more parameter efficient.

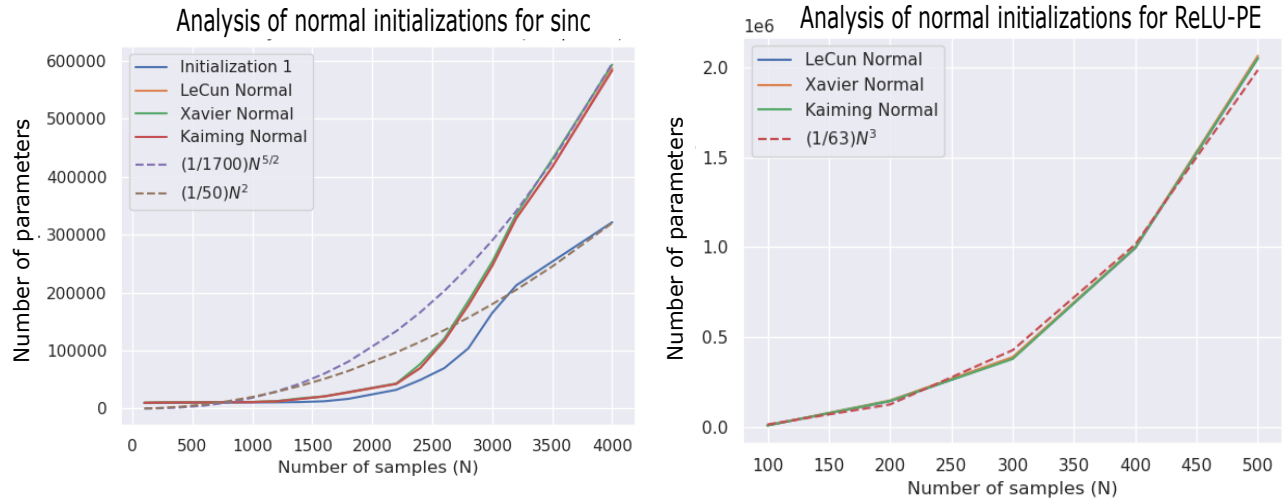

Figure 4. **Left;** Results from initializing four sinc activated networks with different initializations and trained with gradient descent on various image data sample sizes till the networks reach a PSNR of 50dB. As the number of samples go up the networks initialized with LeCun, Kaiming, Xavier Normal initializations all need much more parameter scaling faster than $(1/1700)N^{5/2}$. While the one initialized with initialization 1 needs much less, scaling like $(1/50)N^2$. **Right;** The same experiment as the left though now we measure total parameters of the network and look at ReLU-PE networks. In this case we see that the ReLU-PE networks scaled like $(1/63)N^3$. We thus see that in both experiments initialization 1 with a sinc activated network is far more parameter efficient.

Gradient Descent: For the gradient descent experiments we looked at 6 normal initializations given by the 6 in sec. 2.2. We then also compared the 4 uniform initializations given in sec. 2.2. We used a fixed 4 hidden layer network with each layer having 128 neurons, with a sinc activation with frequency 8. Each network was trained with full batch gradient descent comprising of a total of 38553 points sampled from the Peppers image, see fig. 2. Each network was trained for 20000 epochs. Fig. 6 shows the result. It is clear from this figure that both initialization 1 and 2 perform the best, shwoing that

Figure 5. A $512 \times 512 \times 3$ Lion image for image reconstruction experiments, see sec. 2.4

when parameters are all kept the same these two initializations outperform standard initializations in the literature.

Adam: We decided to also test image reconstruction using the Adam optimizer. For this experiment we used the Lion image, see fig. 5, with a total 150000 sample points, sampled via a Gaussian distribution centred at the origin. We trained 4 sinc networks with the 4 initializations, initialization 1, LeCun Normal, Kaiming Normal and Xavier Normal. We also trained another 4 with the corresponding uniform initializations. All networks had 4 hidden layers and 128 neurons and were trained for 20000 epochs. Fig. 7 shows that in both cases the initializations 1 and 2 completely outperform the others. This validates the generalization of the main theory to the Adam setting carried out in sec. 1.8.

Comparing with SIRENs initialization: We compared initialization 3 against the principle initialization scheme of Sitzmann et al. in [19]. We ran two sine activated network, each with 4 hidden layers, 128 neurons, and the frequency of the sine function set at 30. We then trained both networks on the Peppers image using gradient descent and Adam. Fig. 8 shows that in both cases our initialization 3 outperforms the SIREN initialization by at least 2-3 dB in PSNR. This highlights the ease of our initialization in that it can easily be dropped into many practical initialization schemes.

We ran the above experiment on the Lion instance as well, this time with 150000 samples. We kept everything else the same as in the above experiment on the Peppers image. Fig. 9 shows that in both cases our initialization 3 is able to obtain 2-3 dB higher value in PSNR.

2.5. Computed Tomography (CT) Reconstruction

CT reconstruction is an example of an underconstrained reconstruction problem. We will follow the strategy used in [17] and consider 100 CT measurements of a 256×256 X-ray colorectal image [3].

For this experiment we employed a wavelet activation, see sec. 2.1 for details on the frequency and variance of the wavelet. The goal of this experiment was to test our initialization 1 and 2, see sec. 2.2, against LeCun, Kaiming and Xaiver Normal initializations and against LeCun, Kaiming and Xavier Uniform initializations as these are the most commonly used initializations for neural field applications. In this experiment we kept all parameters the same. Thus all networks employed the same wavelet activation, and had 2 hidden layers each of 300 neurons. The input dimension of the networks was 2 and the output dimension was 1. We used a dataset size of 141810 datapoints and we used the Adam optimizer with full batch training for this task.

We ran two separate experiments. In the first experiment we compared the four normal initializations scheme; LeCun Normal, Kaiming Normal, Xaiver Normal and initialization 1. We then ran a second experiment where we compared LeCun Uniform, Kaiming Uniform, Xaiver Uniform and initialization 2. Fig. 10 shows the results. For normal initializations, initialization 1 achieved at least a 2.5dB higher PSNR and for the uniform initialization, initialization 2 achieved at least a 0.4dB higher PSNR. This validates the extension of the theory to the case of the Adam optimizer as shown in sec. 1.8.

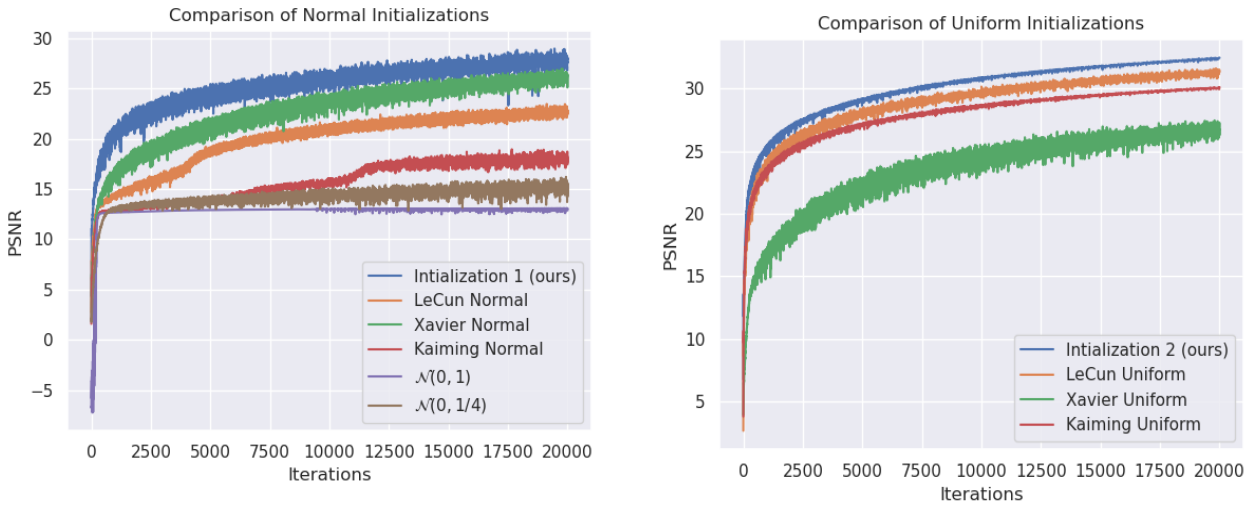

Figure 6. **Left;** Six sinc networks trained with 6 different normal initializations (shown in the legend) on the Peppers image. Initialization 1 achieves at least 2 dB higher than the others. **Left;** Six sinc networks trained with 4 different uniform initializations (shown in the legend) on the Peppers image. Initialization 2 achieves at least 1.3 dB higher than the others.

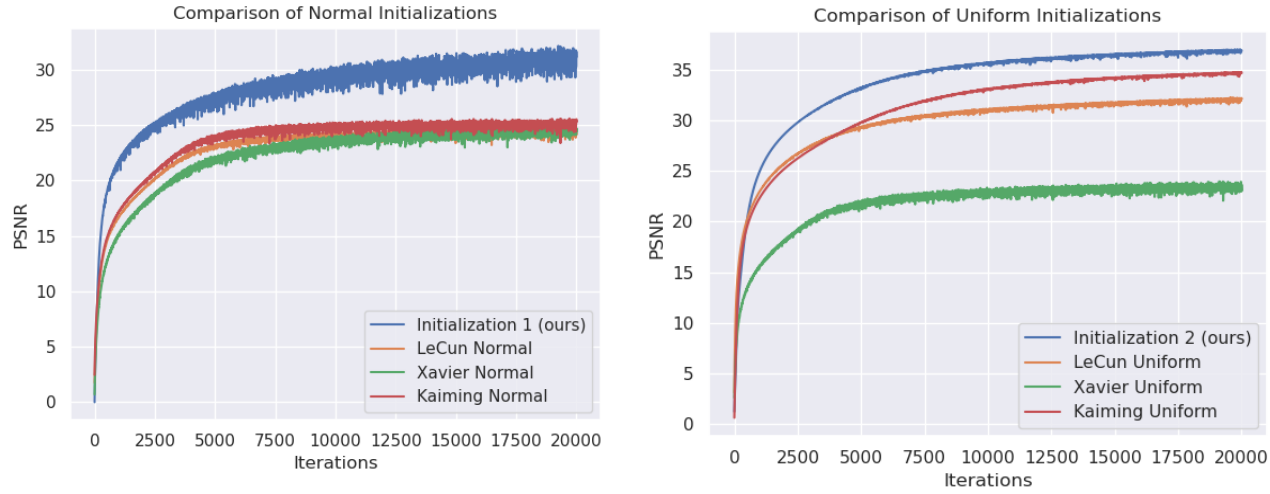

Figure 7. **Left;** Four sinc networks trained with four different normal initializations (shown in the legend) on the Lion image. Initialization 1 achieves at least 5 dB higher than the others. **Left;** Four sinc networks trained with 4 different uniform initializations (shown in the legend) on the Lion image. Initialization 2 achieves at least 3 dB higher than the others.

2.6. Occupancy Fields

In **sec. 5.3** of the main paper we have the results of the occupancy field experiments. Due to space constraints we could not put the reconstructed meshes of each initialization into the main paper. Fig. 11 we show all 8 reconstructions of the meshes with the IOU on the top right. From the figure it is clear that both initializations 1 and 2 outperform all others.

2.7. Neural Radiance Fields

NeRF is commonly trained using uniform initialization schemes and almost always the Kaiming uniform initialization. We decided to run 8 NeRF's, 4 with intialization 1, LeCun Normal, Kaiming Normal, Xavier Normal and another 4 with initialization 2, LeCun Uniform, Kaiming Uniform, Xavier Uniform. We used the Lego instance from the NeRF real synthetic data set. As fig. 12 shows in both cases initialization 1 and initialization 2 outperform the other initializations. For comparison of reconstructions for the unseen scenes, see **sec. 5.4** of the main paper where we compare the two best reconstructions, namely

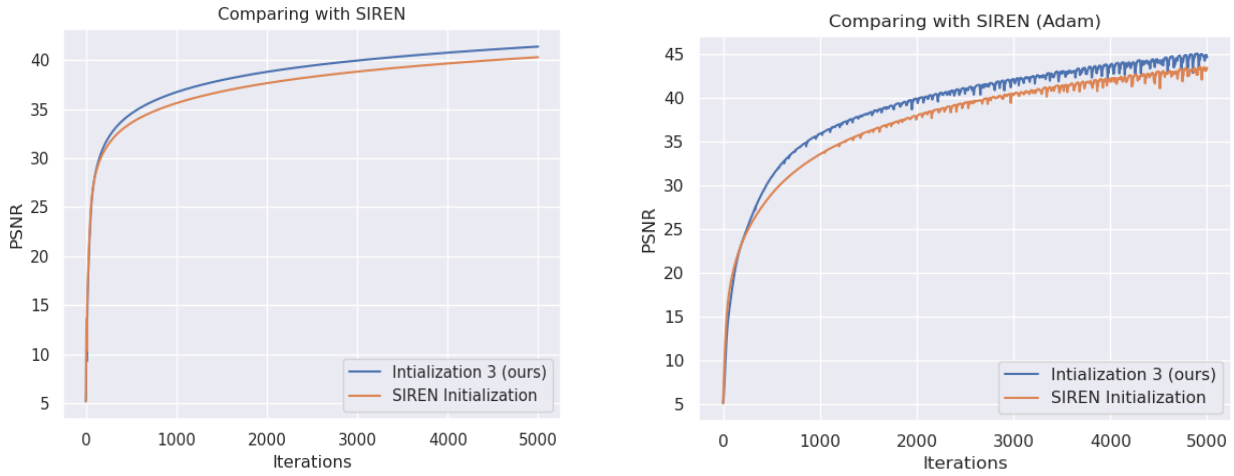

Figure 8. **Left;** Two sine activated networks trained with gradient descent on the Peppers image. Clearly initialization 3 is outperforming SIRENs initialization. **Right;** Two sine activated networks trained with Adam on the Peppers image. Clearly initialization 3 is outperforming SIRENs initialization.

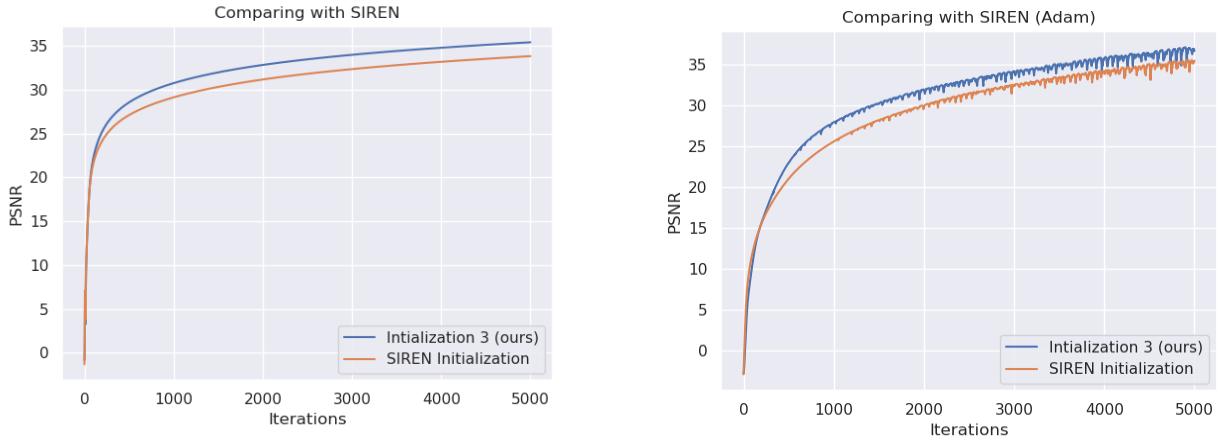

Figure 9. **Left;** Two sine activated networks trained with gradient descent on the Lion image. Clearly initialization 3 is outperforming SIRENs initialization. **Right;** Two sine activated networks trained with Adam on the Lion image. Clearly initialization 3 is outperforming SIRENs initialization.

Kaiming Uniform with Initialization 2.

3. Further Experiments: Applications to physical modelling

In **sec. 5.5** of the main paper we gave results on the Navier-Stokes equations using a physics informed neural network (PINN). For the reader who is unfamiliar with these types of networks, we collect here some basic facts so as to complement **sec. 5.5** of the main paper.

We consider PDEs defined on bounded domains $\Lambda \subseteq \mathbb{R}^n$. To this end, we seek a solution $u : \Lambda \to \mathbb{R}$ of the following system

$$\mathcal{N}[u](x) = f(x), x \in \Lambda \tag{3.1}$$

$$u(x) = f(x), x \in \partial\Lambda. \tag{3.2}$$

where $\mathcal{N}$ denotes a differential operator. In the setting of time-dependent problems, we will treat the time variable t as an

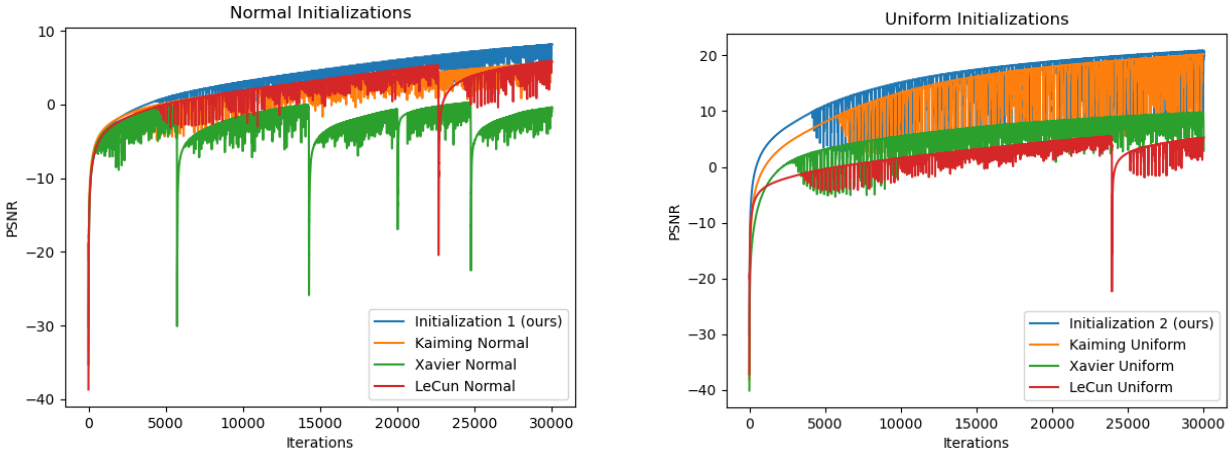

Figure 10. **Left;** Comparison of normal initializations for CT reconstruction task. Initialization 1 (ours) reaches a higher PSNR than all others with a PSNR difference of at least 2.4. **Right;** Comparison of uniform initializations. Initialization 2 (ours) reaches a higher psnr compared to all others with at least 0.4 PSNR difference.

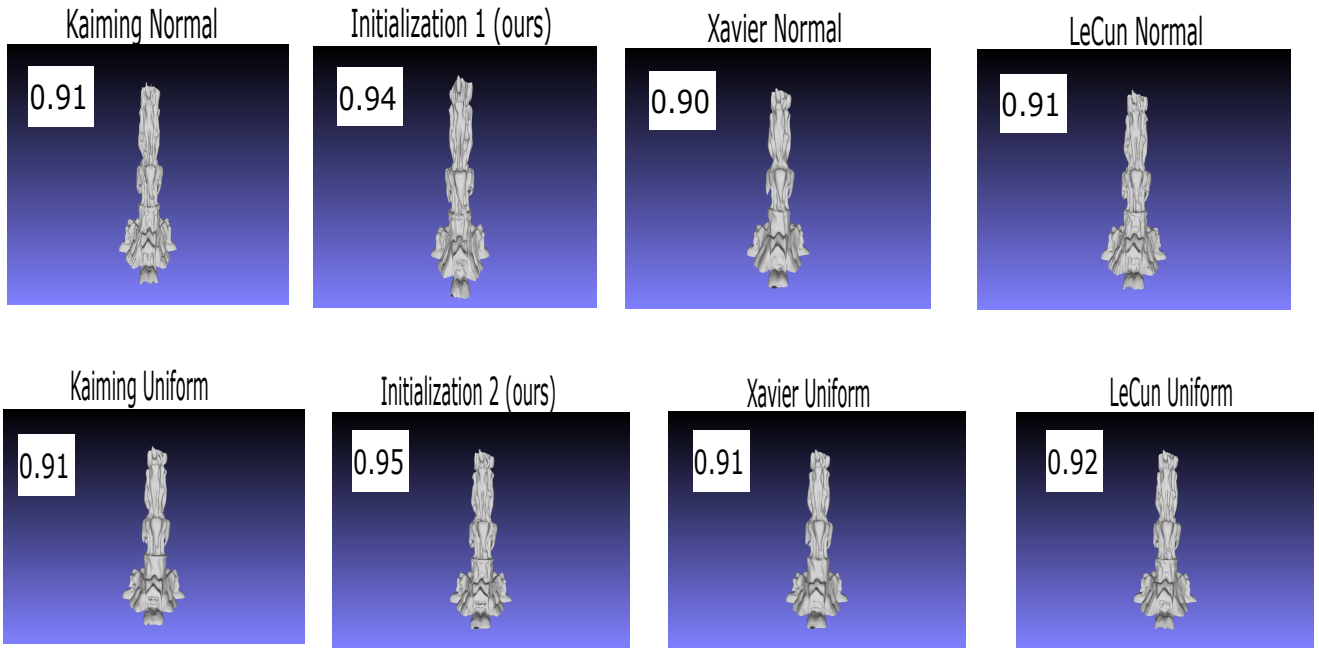

Figure 11. **Top;** Reconstruction of the mesh for all the normal initializations. Initialization 1 has the highest IOU. **Bottom;** Reconstruction of the mesh for all the uniform initializations. Initialization 2 has the highest IOU (zoom in for better viewing).

additional space coordinate and let Λ denote the spatio-temporal domain. In so doing, we are able to treat the initial condition of a time-dependent problem as special type of Dirichlet boundary condition that can be included in (3.2).

The goal of physics informed neural network theory is to approximate the latent solution $u(x)$ of the above system by a neural network $u(x;\theta)$, where θ denotes the parameters of the network. The PDE residual is defined by $r(x;\theta) := u(x;\theta) - f(x)$. The key idea as presented in [14] is that the network parameters can be learned by minimizing the following composite loss

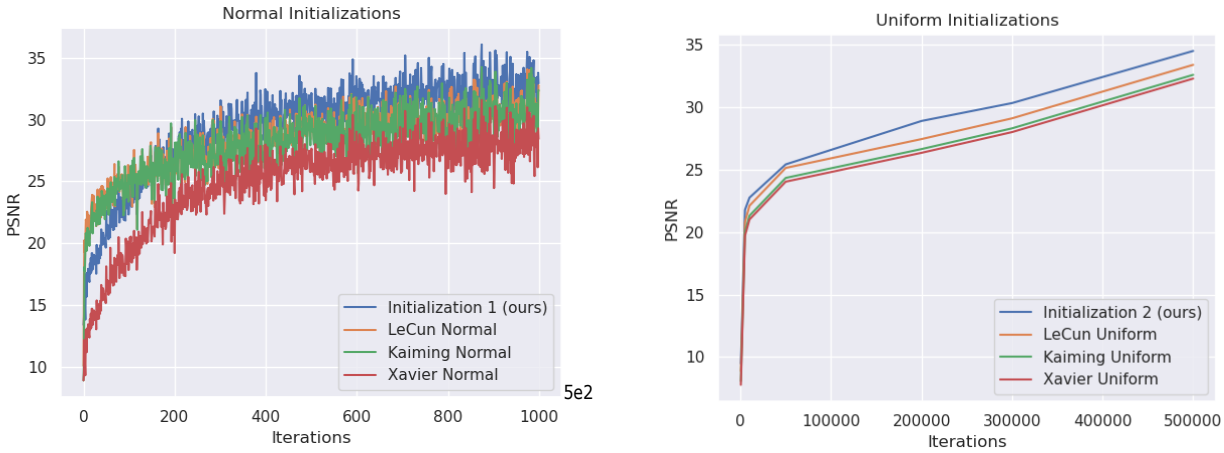

Figure 12. **Left;** Four different NeRF's trained with four different normal initialization schemes. The one trained with initialization 1 reaches at least 0.6dB higher than the others. **Right;** Four different NeRF's trained with four different uniform initialization schemes. The one trained with initialization 2 reaches at least 1.1dB higher than the others.

function

$$\mathcal{L}(\theta) = \mathcal{L}_b(\theta) + \mathcal{L}_r(\theta) \tag{3.3}$$

where $\mathcal{L}_b$ denotes the boundary loss term and $\mathcal{L}_r$ denotes the PDE loss term, defined by

$$\mathcal{L}_b(\theta) = \frac{1}{2N_b}\sum_{i=1}^{N_b} |u(x_b^i;\theta) - g(x_b^i)|^2 \text{ and } \mathcal{L}_r(\theta) = \frac{1}{2N_r}\sum_{i=1}^{N_r} |r(x_r^i;\theta)|^2. \tag{3.4}$$

N_b and N_r represent the training points for the boundary and PDE residual. Minimizing both loss functions, $\mathcal{L}_b$ and $\mathcal{L}_r$, simultaneously using gradient-based optimization aims to learn parameters θ for an effective approximation, $u(x;\theta)$, of the latent solution, see [14]. For the explicit results and experimental setup please see **sec. 5.5** of the main paper.

We consider the 2D incompressible Navier-Stokes equations as considered in [14].

$$u_t + uu_x + 0.01u_y = -p_x + 0.01(u_{xx} + u_{yy}) \tag{3.5}$$
$$v_t + uv_x + 0.01v_y = -p_y + 0.01(v_{xx} + v_{yy}) \tag{3.6}$$

where $u(x, y, t)$ denotes the x-component of the velocity field of the fluid, and $v(x, y, t)$ denotes the y-component of the velocity field. The term $p(t, x, y)$ is the pressure. The domain of the problem is $[-15, 25] \times [-8, 8] \times [0, 20]$. We assume that $u = \psi_y$ and $v = -\psi_x$ for some latent function $\psi(t, x, y)$. With this assumption, the solution we seek will be divergence free, see [14] for details.

References

[1] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019. 21

[2] Shin-Fang Chng, Sameera Ramasinghe, Jamie Sherrah, and Simon Lucey. Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII*, pages 264–280. Springer, 2022. 1, 20

[3] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, et al. The cancer imaging archive (tcia): maintaining and operating a public information repository. *Journal of digital imaging*, 26:1045–1057, 2013. 25

[4] Kenneth R Davidson and Stanislaw J Szarek. Local operator theory, random matrices and banach spaces. *Handbook of the geometry of Banach spaces*, 1(317-366):131, 2001. 16, 17, 18

[5] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019. 21

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 16, 17

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 16, 17, 21

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 19

[9] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 2002. 20

[10] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[11] Quynh Nguyen. On the proof of global convergence of gradient descent for deep relu networks with linear widths. In *International Conference on Machine Learning*, pages 8056–8062. PMLR, 2021. 3

[12] Quynh Nguyen, Marco Mondelli, and Guido F Montufar. Tight bounds on the smallest eigenvalue of the neural tangent kernel for deep relu networks. In *International Conference on Machine Learning*, pages 8119–8129. PMLR, 2021. 16

[13] Quynh N Nguyen and Marco Mondelli. Global convergence of deep networks with one wide layer followed by pyramidal topology. *Advances in Neural Information Processing Systems*, 33:11961–11972, 2020. 10, 14, 15

[14] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707, 2019. 28, 29

[15] S. Ramasinghe and S. Lucey. Beyond Periodicity: Towards a Unifying Framework for Activations in Coordinate-MLPs. In *ECCV*, 2022. 1, 20

[16] Sameera Ramasinghe, Hemanth Saratchandran, Violetta Shevchenko, and Simon Lucey. On the effectiveness of neural priors in modeling dynamical systems. *arXiv preprint arXiv:2303.05728*, 2023. 1, 20

[17] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18507–18516, 2023. 1, 20, 25

[18] Hemanth Saratchandran, Shin-Fang Chng, Sameera Ramasinghe, Lachlan MacDonald, and Simon Lucey. Curvature-aware training for coordinate networks. *arXiv preprint arXiv:2305.08552*, 2023. 1, 20

[19] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, G., and Wetzstein. Implicit Neural Representations with Periodic Activation Functions. In *NIPS*, 2020. 1, 20, 21, 22, 25

[20] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 2

[21] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012. 9

[22] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge university press, 2018. 10, 17, 18