STRONGLY SELF-NORMALIZING NEURAL NETWORKS WITH APPLICATIONS TO IMPLICIT REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies have show that wide neural networks with orthogonal linear layers and Gaussian Poincaré normalized activation functions avoid vanishing and exploding gradients for input vectors with the correct magnitude. This paper introduces a strengthening of the condition that the activation function must be Gaussian Poincaré normalized which creates robustness to deviations from standard normal distribution in the pre-activations, thereby reducing the dependence on the requirement that the network is wide and that the input vector has the correct magnitude. In implicit representation learning this allows the training of deep networks of this type where the linear layers are no longer constrained to be orthogonal linear transformations. Networks of this type can be fitted to a reference image to 1/10th the mean square error achievable with previous methods. Herein is also given an improved positional encoding for implicit representation learning of two-dimensional images and a small-batch training procedure for fitting of neural networks to images which allows fitting in fewer epochs, leading to substantial improvement in training time.

1 INTRODUCTION

Sitzmann et al. (2020) have proposed the use of periodic activation functions, specifically the use of the sine in the fitting of neural networks to, for example, images and signed distance functions. They argue that if the weights are uniformly distributed on $\left[-c/\sqrt{f}, c/\sqrt{f}\right]$ where $c = \sqrt{6}$ and f is the fan in, then the pre-activations are approximately standard normal distributed irrespective of the depth of the network and that the derivatives of these networks are networks of the same type (SIRENs, or sinusoidal representation networks), ensuring that these guarantees are applicable also to derivatives of networks of this type.

This is not the case (see appendix A): if the pre-activations were standard normal distributed, then the activations would have mean zero and variance $\frac{1}{2}\left(1 - e^{-2}\right) \approx 0.43$ and the pre-activations of the next layer mean zero and variance $\frac{1}{6}c^2\left(1 - e^{-2}\right) = 1 - e^{-6} \approx 0.86$.

Furthermore, this provides no guarantees about the derivatives with respect to any parameter.

Such guarantees can be obtained by using higher values of c, indeed, with sufficiently high values of c it can be assured (see appendix A) that if the pre-activations are $\mathcal{N}(0, c^2/6)$-distributed then the pre-activations of the next layer will be as well and that if the gradient with respect to one layer has become $\mathcal{N}(0, c^2/6)$-distributed, then the pre-activation of the layer before it will also be $\mathcal{N}(0, c^2/6)$-distributed.

When training networks with large constant learning rates it is not immediately apparent that these guarantees are of any benefit, in appendix A there are training curves from which it is apparent that SIREN networks with $c = \sqrt{6}$ learn faster than when larger values of c are used during the portion of training before learning rates are reduced.

However, when appropriate training procedures are used it is possible to benefit from the guarantees obtainable in the case of SIREN networks with larger c. It is possible to fit a SIREN with width 256 and five hidden layers to the 512×512 Cameraman test image used for the same purpose inSitzmann

et al. (2020) and achieve a PSNR of 57.5, provided that an initial learning rate of $5 \cdot 10^{-4}$ is used with a learning rate schedule involving the halving of the learning rate when the MSE plateaus for 60 epochs. SIRENs with $c = \sqrt{6}$ can with the same training method achieve PSNR 52.99. The method used bySitzmann et al. (2020) for fitting SIRENs to images involves a constant learning rate of $1\dot{1}0^{-4}$ and 15,000 epochs and full-batch training and achieves a PSNR of approximately 50.

Lu et al. (2020) have proposed a type of neural network, bidirectionally self-normalizing neural networks (BSNNs), for which they prove similar guarantees as those available for large-c SIRENs: provided that the layers are orthogonal linear transformations that are uniformly distributed on the orthogonal group in the Haar sense followed by activation functions that are Gaussian-Poincaré normalized, meaning that the activation function f satisfies $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[f(z)^2] = 1$ and $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[\frac{df}{dx}(z)^2] = 1$; f and its derivative are Lipschitz continuous, and the input vector is thin-shell concentrated in the sense that for all $\epsilon > 0$ $\mathbb{P}\left\{|\frac{1}{n}\|x\|^2 - 1| > \epsilon\right\} \to 0$ as $n \to \infty$ and that the layers are wide it can assured that the squared magnitude of the input to each layer is n and the magnitude of the derivative of any loss function E with respect to the input to any layer is the same.

The guarantee that a thin-shell concentrated vector has its norm preserved under forward propagation in a BSNN is comparable to the guarantee that in a SIREN with sufficiently large c the pre-activations all have the distribution, while the guarantee that the derivative of a loss function with respect to the input to any layer always has the same magnitude corresponds to the assurance provided in appendix A for SIRENs with sufficient large c that the backward propagation, once the gradient is approximately $\mathcal{N}(0, c^2/6)$-distributed the gradient to earlier layer will be as well.

We consider a condition under which a BSNN with orthogonal initialization but no orthogonality constraint performs well in implicit representation learning. Among the BSNNs satisfying this condition is one all of whose derivatives are networks of the same type.

2 MOTIVATION

In a BSNN of some fixed finite width the pre-activations are uniformly distributed on the sphere with radius equal to the square root of the dimension, which in high dimension in a certain sense closely approximates the normal distribution so that the trainability guarantees of (Lu et al., 2020) can be obtained about the distribution of the activations even though the condition they impose is on the expectation of the activation function and its derivatives applied to a standard normal distributed random variable.

It is possible to impose a stronger condition, that $\mathbb{E}_{z\sim W}[f(z)^2] = \mathbb{E}_{z\sim W}[\frac{df}{dx}(z)^2] = 1$ for all distributions W such that $W = -W$. This is the case precisely if $f(x)^2 - 1$ and $\frac{df}{dx}(x)^2 - 1$ are odd functions.

We consider the ideal case where vectors with magnitude exactly equal to the square root of dimension are forward propagated through neural networks of different width with different activation functions, all of which are such that $\mathbb{E}_{z\sim\mathcal{N}(0,1)}[f(Z)^2] = 1$ and some such that $f(x)^2 - 1$ is an odd function.

In fig. 1. a measure of deviation of the magnitude of the pre-activations of BSNNs from the square root of the dimension is calculated for different activation functions and network widths. The activation functions giving the the two lowest values of this measure measure (the curves marked gold and black) both satisfy the stronger condition (although that achieving the lowest average, $\sqrt{2/(1 + e^{-x})}$ without satisfying the part of the Gaussian Poincaré normalization condition that concerns the derivativ so that the guarantees of Lu et al. (2020) with regard to backward propagation do not hold and so that that function is not usable for our purposes).

Had we looked, instead of at deviations of the magntiude from the square root the dimension, at ratios of magnitude of output layers to input layers, we would see no benefit of activation functions satisfying the stronger condition.

What the stronger condition assures is not that the magnitudes are preserved more exactly, but that the magnitude of a vector whose magnitude deviates from the square root of the dimension is brought closer to that magnitude. This can be see in fig. 2. showing a measure of the deviation of the magntiude of the pre-activations from the square root of the dimension where the initial input vector

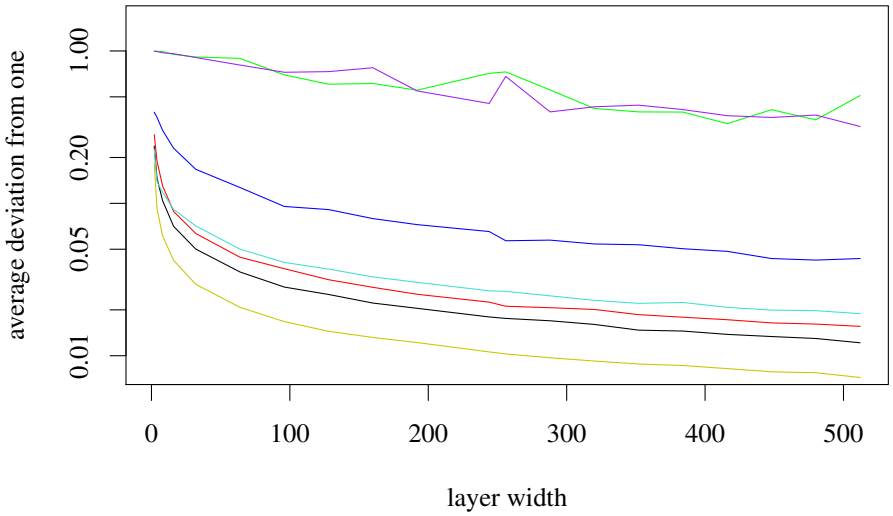


Figure 1: Log plot of $\sum_k |\frac{\|x_k\|}{\sqrt{n}} - 1|$ where x_k is the pre activations over 400 layers, averaged over ten runs, as a function of the network width. Gold: $\sqrt{2/(1+e^{-x})}$, Black: $\sqrt{2}\sin(x+\pi/4)$, Red: GP normalized $\tanh$, Turquoise: GP normalized GELU, Blue: GP normalized ELU, Purple: GP normalized leaky ReLU, Green: GP normalized ReLU.

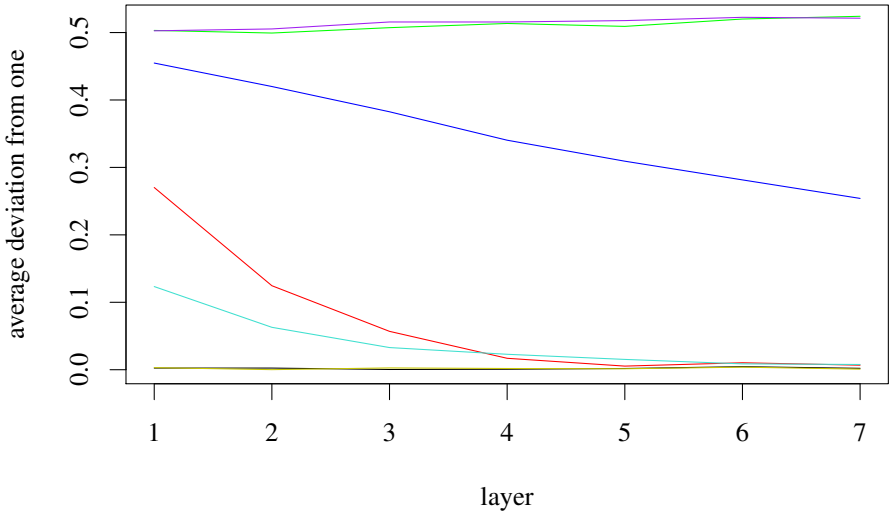


Figure 2: $|\frac{\frac{1}{100}\sum_{l=1}^{100}\|x_k^{(l)}\|}{\sqrt{n}} - 1|$ for k=1,..,7 where $x_k^{(l)}$ are the pre-activations for layer k during forward propagation of $x_0^{(l)}$ and $x_0^{(l)}$ is a random normal vector transformed by normalization to have norm $\frac{\sqrt{n}}{2}$ where n is the dimension. Colour corresponds to activation function. Gold: $\sqrt{2/(1+e^{-x})}$, Black: $\sqrt{2}\sin(x+\pi/4)$, Red: GP normalized $\tanh$, Turquoise: GP normalized GELU, Blue: GP normalized ELU, Purple: GP normalized leaky ReLU, Green: GP normalized ReLU.

magnitude of $\frac{1}{2}$ the square root of the dimension. For some of the activation functions the deviation tends to zero, but in the case of the activation function satisfying the stronger condition, they are zero immediately.

We will be concerned with relaxations of networks of the following type:

Definition 1 (Strongly bidirectionally self-normalizing neural networks)**.** *A neural network is said to be a SBSNN if it has orthogonal weight matrices that are uniformly distributed in the Haar sense*

and the activation is differentiable, Lipschitz continuous with Lipschitz continuous derivative and has the property that $f(x)^2 - 1$ and $\frac{df}{dx}(x)^2 - 1$ are odd functions.

3 CHARACTERIZATION OF SBSNNS

Lemma 1. *If $f(x)^2 - 1$ and $\frac{df}{dx}(x)^2 - 1$ are odd functions then there is an even function w taking values in $\{1, -1\}$ such that $f(x)^2 - 1 = \sin(2\int_0^x w(s)ds)$.*

Proof. Let $u(x) = f(x)^2 - 1$ and $v(x) = \frac{df}{dx}(x)^2 - 1$. These are then odd functions.

Because $v(x) = -v(-x)$ we see that $\frac{df}{dx}(x)^2 - 1 = -(\frac{df}{dx}(-x)^2 - 1)$ so that $\frac{df}{dx}(x)^2 + \frac{df}{dx}(-x)^2 = 2$.

Also, $\frac{du}{dx}(x) = 2f(x)\frac{df}{dx}(x)$ so that $\frac{du}{dx}(x)^2 = 4f(x)^2\frac{df}{dx}(x)^2$. Because $u(x) = f(x)^2 - 1$ we can write $f(x)^2 = u(x) + 1$ and in turn that $\frac{du}{dx}(x)^2 = 4(u(x) + 1)\frac{df}{dx}(x)^2$.

We now multiply $\frac{df}{dx}(x)^2 + f(-x)^2 = 2$ by $4(u(x) + 1)\frac{df}{dx}(x)^2 \cdot 4(u(-x) + 1)\frac{df}{dx}(-x)^2$ and simplifying using the definition of $\frac{du}{dx}$ we obtain that $4(u(-x)+1)\frac{du}{dx}(x)^2 + 4(u(-x)+1)^2\frac{du}{dx}(x)^2 = 2 \cdot 4(u(x) + 1) \cdot 4(u(-x) + 1)$.

Further simplifying and using that u is an odd function we obtain that $4\frac{du}{dx}(x)^2(u(x) + 1 - u(x) + 1) = 2 \cdot 4^2(1 - u(x)^2)$. Further simplifying we obtain $\frac{du}{dx}(x)^2 = 4(1 - u(x)^2)$. By definition $u(x) > -1$. Suppose that $u(x) > 1$ then $u(-x) = -u(x) < -1$, which is impossible. Consequently $|u(x)| \leq 1$. Taking the square root of $1 - u(x)^2$ is thus permissible and we obtain $|\frac{du}{dx}(x)| = 2\sqrt{1 - u(x)^2}$.

Let $\bar{w}(x) = \text{sgn}(\frac{du}{dx}(x))$. Since $\frac{du}{dx}$ is the derivative of an odd function it is even and $\bar{w}$ is even.

Now $\frac{du}{dx}(x) = 2w(x)\sqrt{1 - u(x)^2}$, consequently $\frac{du}{\sqrt{1-u(x)^2}} = 2\bar{w}(x)dx$ and $\arcsin(x) + C = 2\int_0^x \bar{w}(s)ds$ for some constant C. Thus $u(x) = \sin(2\int_0^x \bar{w}(s)ds - C)$. Because u is odd it is necessary that $u(0) = 0$ and thus that $C = \pi \cdot n$. Since $\sin(x + \pi) = -\sin(x)$ for all x $u(x) = \sin(2S\int_0^x \bar{w}(s)ds)$ with $S \in \{1, -1\}$ and by setting $w(x) = S\bar{w}(x)$ the conclusion holds. □

Theorem 1. *f is an activation function of an SBSNN, i.e. such that f is differentiable, Lipschitz continuous, has a Lipschitz continuous derivative and satisfies that $f(x)^2 - 1$ and $\frac{df}{dx}(x)^2 - 1$ are odd functions, precisely if $f(x) = \pm\sqrt{2}\sin(x + \pi/4)$ or $f(x) = \pm\sqrt{2}\cos(x + \pi/4)$.*

Proof. By lemma 1 there is an even function w taking values in $\{1, -1\}$ such that $f(x)^2 - 1 = \sin(2\int_0^x w(s)ds)$. Let $I(x) = \int_0^x w(s)ds$. Then $f(x)^2 = 1 + \sin(2I(x)) = \cos(I(x))^2 + \sin(I(x))^2 + 2\sin(I(x))\cos(I(x)) = (\cos(I(x)) + \sin(I(x)))^2 = (\sin(I(x) + \pi/4))^2$. Thus there exists some function $S(x)$ such that $f(x) = S(x)\sqrt{2}\sin(\int_0^x w(s)ds + \pi/4)$ where $S(x)$ takes values in $\{1, -1\}$.

Since S takes values in $\{1, -1\}$ and $\sqrt{2}\sin(\int_0^x w(s)ds + \pi/4)$ is continuous, if S jumps at any input where $\sin(\int_0^x w(s)ds + \pi/4)$ is not zero, then $f(x)$ is not continuous. Consequently $S(x)$ changes sign only at points where $\sin(\int_0^x w(s)ds + \pi/4) = 0$.

Thus at points where $\sin(\int_0^x w(s)ds) \neq 0$ $f'(x) = S(x)\sqrt{2}\cos(\int_0^x w(s)ds + \pi/4)w(x)$. Continuity of $\frac{df}{dx}(x)$ then requires that $S(x)$ jumps precisely where $w(x)$ jumps.

Thus $w(x)$ and $S(x)$ may only jump at point where $\sin(\int_0^x w(s)ds + \pi/4) = 0$ and if they jump at such a point they must jump together. Consequently we can write $S(x) = Cw(x)$ where $C \in \{1, -1\}$. Thus $\frac{df}{dx}(x) = \sqrt{2}Cw(x)\cos(\int_0^x w(s)ds + \pi/4)$. If w jumps at x, then since $\int_0^x w(s)ds = 0$ it follows that $\cos(\int_0^x w(s)ds) = 1$, so $\frac{df}{dx}$ jumps at x, and $\frac{df}{dx}$ is not continuous, which is a contradiction.

Thus w is constant and $f(x) = C\sin(Wx + \pi/4)$ where $C, W \in \{1, -1\}$.

The reverse direction is trivial. □

As a corollary of this result it follows that all derivatives of SBSNNs are in turn SBSNNs.

Lipschitz continuity of derivatives and the activation function and its derivative is a requirement of the theory of Lu et al. (2020), but Gaussian Poincaré normalized ReLU activation functions, which do not have a derivative everywhere and which have a derivative which is not Lipschitz continuous still behave largely according to the theory, with wide Gaussian Poincaré normalized ReLU networks preserving norms in practice (fig. 1). Consequently there is reason to relax the continuity conditions so as to admit functions for which the guarantees of Lu et al. (2020) do not hold but which are still in the spirit of the theory. For this reason we consider the following theorem:

Theorem 2. *Let w be an even function taking values in $\{1, -1\}$ such that $\partial \int_0^x w(s)ds$ exists where ∂ is a left- or right derivative and S any function taking values in $\{1, -1\}$ with jumps only when $\sin(\int_0^x w(s)ds + \pi/4) = 0$, then $f(x) = \sqrt{2}S(x)\sin(\int_0^x w(s)ds + \pi/4)$ is such that $f(x)^2 - 1$ and $(\partial f)(x)^2 - 1$ are odd functions.*

Proof. Let as before $I(x) = \int_0^x w(s)ds$. Then $I(x)$ is odd.

$f(x)^2 - 1 = 2\sin^2(I(x) + \pi/4) - 1 = 2\left(\frac{1}{\sqrt{2}}\left(\sin(I(x)) + \cos(I(x))\right)\right)^2 - 1 =$
$= 1 + 2\sin(I(x))\cos(I(x)) - 1 = 2\sin(I(x))\cos(I(x))$ is an odd function since $I(x)$ and $\sin(x)\cos(x)$ are odd functions.

Let $g(x) = S(x)\sqrt{2}\sin(x+\pi/4)$. Because $S(x)$ jumps only when $\sin(I(x)+\pi/4)$ is zero ∂g exists and is $\lim_{a\uparrow x} S(a)\sqrt{2}\cos(x+\pi/4)$ in the case of the left derivative and $\lim_{a\downarrow x} S(a)\sqrt{2}\cos(x+\pi/4)$ in the case of the right derivative.

Similarly, since $(\partial I)(x)$ exists it is $\lim_{a\uparrow x} w(a)$ in the case when ∂ is the left derivative and $\lim_{a\downarrow x} w(a)$ in the case of the right derivative.

Consequently $\partial f = \partial(g \circ I)$ exists and is $(\partial g)(I(x))(\partial I)(x)$. $(\partial I)(x)$ is in $\{1, -1\}$ so $((\partial f)(x))^2 = (\partial g)(I(x))^2 = (\lim_{a\uparrow x} S(a)\sqrt{2}\cos(I(x) + \pi/4))^2 = 2\cos^2(I(x) + \pi/4)$.

Consequently $(\partial f)(x)^2 - 1 = 2\cos^2(I(x) + \pi/4)\bar{w}(x)^2 - 1 =$
$= 2(\frac{1}{\sqrt{2}}(\cos(I(x)) - \sin(I(x)))^2 - 1 = -2\cos(I(x))\sin(I(x))$, which is an odd function. □

4 POSITIONAL ENCODERS

SBSNNs require input which has magnitude equal to the square root of the dimension. Thus effective use requires the use of a positional encoder.

Mildenhall et al. describe a positional encoder assigning to each co-ordinate the vector $(\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p))$ where p is the co-ordinate. These vectors have squared magnitude equal to one half their dimension, and by scaling them by $\sqrt{2}$ we obtain a vector of the required magnitude.

The method of Mildenhall et al. was originally applied to five-dimensional input, but when applied to two-dimensional input it introduces obvious patterns in the form of correlations between pixels along lines where one co-ordinate is constant (see fig. 3).

These patterns can be removed by a slight modification of the encoder of Mildenhall et al.: taking the input co-ordinate pair $(x, y)^T$ we construct two additional co-ordinate pairs by rotating the original co-ordinate pair by one third and two-thirds a turn around the origin to obtain two more $D_{2\pi/3}(x, y)^T$, $D_{4\pi/3}(x, y)^T$ where D_θ denotes a rotation in the x-y plane by θ, and applying the encoder of Mildenhall et al. to each and concatenating the three outputs. Use of this positional encoder avoids line artefacts early (fig. 3) in the network fitting and can be seen to improve final mean squared error.

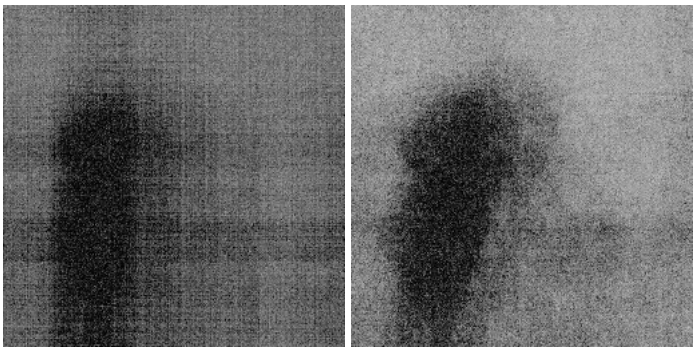


Figure 3: Early frame during fitting. Left: Using a scaled encoder of the Mildenhall et al. type, Right: Using the modified encoder. The image being fitted is a scaled-down 256×256 version of the Cameraman test image.

Table 1: Comparison of image fitting of the 512×512 Cameraman test image using batch size 256 an initial learning rate of $5 \cdot 10^{-4}$ and with a halving of the learning rate when MSE plateaus for 60 epochs.

ARCHITECTURE	**FINAL PSNR**
SIREN, $c = 5.1$	56.2
SIREN, $c = \sqrt{6}$	53.2
One SIREN layer followed by SBSNN	55.40
Mildenhall encoder followed by SBSNN	66.9
Rotated encoder followed by SBSNN	67.53

5 Experimental results

Results of image fitting experiments are summarized in table 1. Counting the PSNR 56.2 as what is achievable with previous methods claim of the abstract follows: a PSNR of 56.2 corresponds to an MSE of $10^{-56.2/10} = 2.39 \cdot 10^{-6}$ and a PSNR of 67.53 to an MSE of $10^{-67.53/10} = 1.7 \cdot 10^{-7}$. Thus the MSE achieved by this method is less than 1/10th that achievable previous methods.

With regard to training time, it is straightforward to see that this method uses substantially less computation, as it is able to fit an image to superior accuracy in 500 epochs, instead of 15000 epochs. Even so, on large GPUs full batch training will make more effective use of the machine. In the these experiments however, full batch training took 200 ms per epoch, while training with batch size 256 took 2048 ms per epoch. Thus total training time is 3000 seconds for the traditional method vs 1024 seconds with the improved method.

References

Y. Lu, Gould. S., and Ajanthan T. Bidirectionally self-normalizing neural networks. *arXiv preprint arXiv:2006.12169*, 2020.

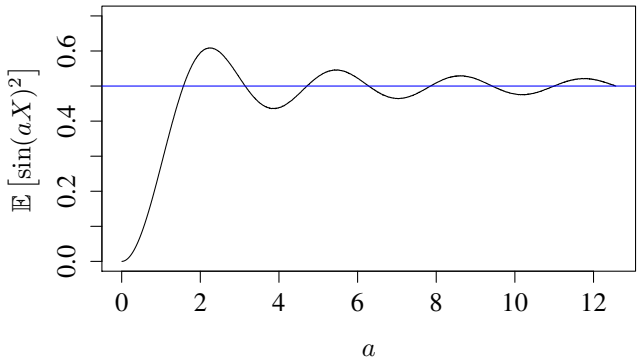
B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, Ramamoorthi R., and Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. *Computer Vision – ECCV 2020, 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I*, pp. 405–421.

V. Sitzmann, J.N.P. Martel, A.W. Bergman, and D.B. Lindell. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

A APPENDIX

A.1 FORWARD PROPAGATION IN SIRENS

In relation to the case where the pre-activations are uniformly distributed it has been claimed that when $X \sim \mathcal{U}(-1, 1)$ $\sin(aX + b)$ will be Arcsine$(-1, 1)$ distributed irrespective of b. This is only approximately true. The variance of a Arcsine$(-1, 1)$ distribution is $1/2$. There is a change in behaviour at $\pi/2$, where the variance finally reaches $1/2$ and for values greater than $\pi/2$ the variance remains close to $1/2$, but it is in fact not $1/2$ in general and the difference is not altogether small, unless c is large, as can be seen from the graph below. The blue line shows $y = 1/2$. This can be resolved either by scaling the initial uniform distribution, thus causing the input to be the $\mathcal{U}[-\pi/2, \pi/2]$ distributed instead of $\mathcal{U}[-1, 1]$, by using a precisely scaled activation function, $\sin(a\cdot)$ where $a = \frac{\pi}{2}$ instead of $\sin(\cdot)$, or by using a heavily scaled activation function $\sin(a\cdot)$ where a might be 30. This last approach is what has been proposed by Sitzmann et al. (2020) for use in practice, while they use $a = \pi/2$ in a theoretical analysis.



The normal distribution of the pre-activations that occurs in the layers following the first arises as follows:

The activations from the previous layer are assumed to be Arcsine$(-1, 1)$ distributed and because of this they have mean zero and variance $1/2$.

The synaptic weights are independent of the activations of the previous layer and are $\mathcal{U}(-c/\sqrt{n}, c/\sqrt{n})$ distributed. This distribution has mean zero and variance $\frac{1}{12}(2c/\sqrt{n})^2 = \frac{1}{3}c^2/n$.

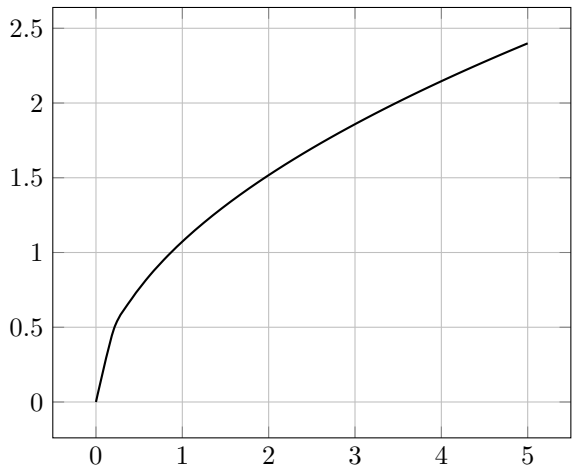
The pre-activations for the new layer are the dot product of the previous activations and the synaptic weights of the new layer. The mean of the this is zero and the variance of the product of a single activation X_i and its corresponding synaptic weight W_i is

$$\begin{aligned}
\text{Var}[W_i X_i] &= \mathbb{E}\left[(W_i X_i - E[W_i X_i])^2\right] = \\
&= \mathbb{E}\left[W_i^2 X_i^2 - 2 W_i X_i \mathbb{E}\left[W_i X_i\right] + E[W_i X_i]^2\right] = \\
&= \mathbb{E}\left[W_i^2\right] \mathbb{E}\left[X_i^2\right] - \mathbb{E}\left[W_i\right]^2 \mathbb{E}\left[X_i\right]^2 = \\
&= \frac{1}{3}c^2/n \cdot \frac{1}{2} - 0 \cdot 0 = \frac{1}{6}c^2/n
\end{aligned}$$

We know now that $\sqrt{n} W_i X_i$ are independent, identically distributed random variables with mean zero and variance $\frac{1}{6}c^2$ and thus, by the central limit theorem $\frac{1}{\sqrt{n}} \sum_{i=1}^{n} \sqrt{n} W_i X_i \xrightarrow{d} N\left(0, \frac{1}{6}c^2\right)$. Thus $\sum_{i=1}^{n} W_i X \xrightarrow{d} N(0, \frac{1}{6}c^2)$.

It has been claimed that $\sin(W^T X)$ will be Arcsin$(-1, 1)$ if $c > \sqrt{6}$. This is not the case: $\text{Var}[\sin(kZ)] = \frac{1}{2}(1 - e^{-2k^2})$. Consequently, when $c = \sqrt{6}$ $W^T X \sim N(0, 1)$, giving $\text{Var}[\sin(Z)] = \frac{1}{2}(1 - e^{-2}) \approx 0.43$, but if $\sin(Z)$ had been Arcsin$(-1, 1)$ it would have variance $1/2$.

The k at which $\frac{1}{2}(1 - e^{-2k^2}) = 0.5 - \epsilon$ does not grow quickly when ϵ is made small. The k which produces $\epsilon = 10^{-p}$ is $k = \sqrt{p \log(10)/2}$.



This gives a reason to choose higher values of c than $\sqrt{6}$ or $\frac{\pi}{2\sqrt{6}}$. Another will come from the following analysis of backward propagation in SIRENs.

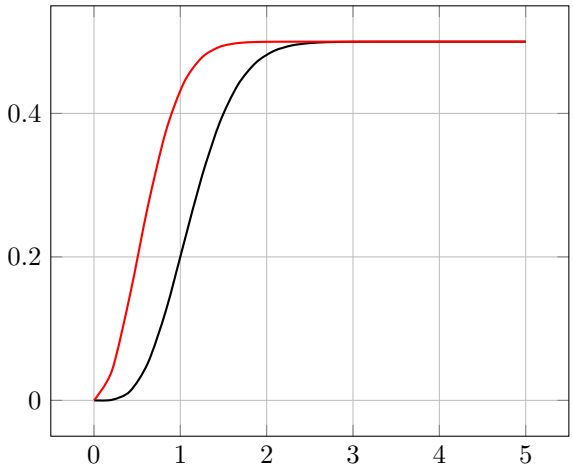
A.2 Backward propagation in SIRENs

Consider the elementwise nonlinearity of a SIREN neuron and the set of synapses which receive input from it. This is a function $f : \mathbb{R} \to \mathbb{R}^n$ assigning to an input x the vector $(w_1 \sin(x), ..., w_n \sin(x))^T$. Consider the case when $\frac{\partial E}{\partial f(x)_k}$ are known. Then $\frac{\partial E}{\partial x} = \frac{\partial E}{\partial f(x)_k} \frac{\partial f(x)_k}{\partial x} = \sum_k \frac{\partial E}{\partial f(x)_k} w_k \cos(x)$.

Treating the gradients $\frac{\partial E}{\partial f(x)_k}$ as inputs and the gradient $\frac{\partial E}{\partial x}$ we obtain a dual neuron with activation function $\cos(\cdot)$ and weights w_k. The weight distribution of $(w_k)_{k=1}^n$ will be the same as for forward layers.

Over many fully connected layers their pre-activations, i.e. gradients before the dual activation function is applied will become approximately normal distributed: when the input distribution has high variance it will be approximately uniformly distributed on a wide interval, and such a distribution transformed by the cosine will be approximately Arcsine$(-1, 1)$ distributed. Consequently this will lead to approximately normal distributed pre-activations.

However, for the cosine of zero-centred normal distribution to be approximately Arcsine$(-1, 1)$ this distribution must have higher variance than for the sine of the same distribution to be Arcsine$(-1, 1)$ distributed. This can be seen by considering the variance $\text{Var}[\cos(kZ)] = \frac{1}{2} + \frac{1}{2}e^{-2k^2} - e^{-k^2}$ to the variance $\text{Var}[\sin(kZ)] = \frac{1}{2}(1 - e^{-2k^2})$.

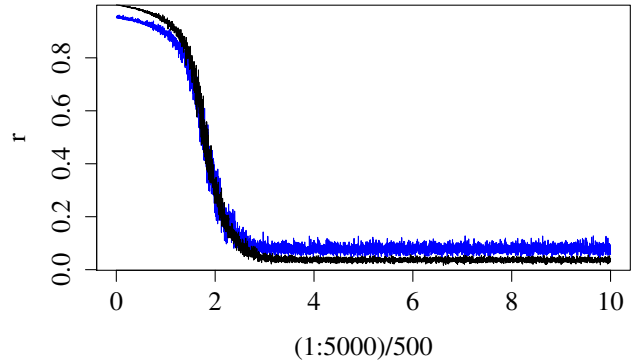


The black curve shows the variance of the cosine transformed zero-centred normal distribution and of the sine-transformed zero-centred normal distribution as a function of the standard deviation of the input distribution.

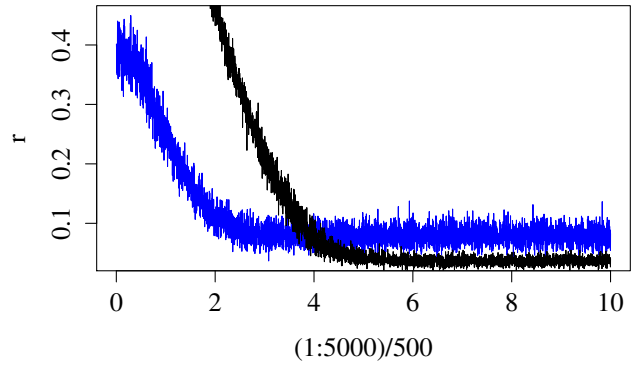
A.3 SIMULATIONS

The need for higher variance pre-activations becomes apparent in simulations. We consider a SIREN neural network with 20 layers and width 64 which receives $\mathcal{U}[-\pi/2, \pi/2]$ at initialization, for initializations with different choices of c.

The blue and black curves show, respectively, the mean of the absolute deviation of the pre-activations and of the activations from their intended values, $c/\sqrt{6}$ for the pre-activations and $\sqrt{1/2}$ for the activations, divided by those intended values, as a function of c.



In this we show a dual network corresponding to gradient propagation, unrealistically receiving $\mathcal{U}[-\pi/2, \pi/2]$ distributed input.



This gives reason to consider values of c as large as 5.

B DERIVATIONS OF FORMULAS

B.1 DERIVATION OF THE FORMULA FOR $\boldsymbol{E[\sin(kZ)^2]}$

We give the derivation of the formula for $\mathbb{E}[\sin(kZ)^2]$. Consider a mean of a particular transformed Wiener process

$$f(t) = \mathbb{E}\left[\sin^2(cW_t)\right].$$

Knowing the derivatives

$$\frac{d}{dx}\sin^2(cx) = 2c\sin(cx)\cos(cx) = c\sin(2cx)$$

.

$$\frac{d^2}{dx^2}\sin(x)^2 = 2c^2\cos(2cx),$$

we may apply Itô's lemma

$$\begin{aligned}\sin^2(cW_t) &= \int_0^t c\sin(2cW_s)dW_s + \frac{1}{2}\int_0^t 2c^2\cos(2cW_s)ds = \\ &= \int_0^t c\sin(2cW_s)dW_s + \int_0^t c^2\cos(2cW_s)ds.\end{aligned}$$

Consequently

$$f(t) = c^2\int_0^t \mathbb{E}\left[\cos(2cW_s)\right]ds.$$

Now, consider

$$g(t,u) = \mathbb{E}\left[\cos(uW_t)\right].$$

Applying Itô's lemma we obtain

$$\begin{aligned}g(t,u) = \mathbb{E}\left[\cos(u(W_s))\right] = \mathbb{E}\left[\frac{1}{2}\int_0^t -u^2\cos(uW_s)ds\right] = \\ = -\frac{1}{2}u^2\int_0^t \mathbb{E}\left[\cos(uW_s)\right]ds = -\frac{1}{2}u^2\int_0^t g(s,u)ds.\end{aligned}$$

Thus

$$\frac{\partial g}{\partial t}(t,u) = -\frac{1}{2}u^2 g(t,u)$$

and

$$g(t,u) = C(u)e^{-\frac{1}{2}u^2 t}$$

.

Since

$$g(0,u) = 1$$

We have

$$C(u) = 1$$

and thus that

$$g(t,u) = e^{-\frac{1}{2}u^2 t}.$$

Thus

$$\begin{aligned} f(t) = c^2 \int_0^t \mathbb{E}\left[\cos(2cW_s)\right] ds = c^2 \int_0^t g(t,2c) = \\ = c^2 \int_0^t e^{-\frac{1}{2}4c^2 t} = c^2 \int_0^t e^{-2c^2 t} = \\ = c^2 \frac{e^{-2tc^2} - 1}{-2c^2} = \frac{1 - e^{-2c^2 t}}{2}. \end{aligned}$$

We can conclude that

$$\mathbb{E}\left[\sin(cZ)^2\right] = \frac{1 - e^{-2c^2}}{2}.$$

B.2 Derivation of the formula for Var $[\cos(cZ)]$

In the calculation of $\mathbb{E}[\sin(cZ)^2]$ we obtained two results that are relevant also for this calculation, that

$$\mathbb{E}[\cos(uW_t)] = e^{-\frac{1}{2}u^2 t}$$

and the conclusion of the previous calculation, that

$$\mathbb{E}[\sin(cZ)^2] = \frac{1 - e^{-2c^2}}{2}.$$

Because

$$\mathbb{E}\left[\sin(cZ)^2 + \cos(cZ)^2\right] = 1$$

it immediately follows that

$$\begin{aligned} \mathbb{E}\left[\cos(cZ)^2\right] = 1 - \mathbb{E}\left[\sin(cZ)^2\right] = \\ = 1 - \frac{1 - e^{-2c^2}}{2} = \frac{1}{2} + \frac{1}{2}e^{-2c^2}. \end{aligned}$$

Using that

$$\text{Var}\left[X\right] = \mathbb{E}\left[\left(X - \mathbb{E}\left[X\right]\right)^2\right] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

we obtain

$$\begin{aligned}\text{Var}\left[\cos\left(cZ\right)\right] &= \frac{1}{2} + \frac{1}{2}e^{-2c^2} - \left(e^{-\frac{1}{2}c^2}\right)^2 = \\ &= \frac{1}{2} + \frac{1}{2}e^{-2c^2} - e^{-c^2}\end{aligned}$$