Neural Networks, All of Whose Derivatives are Bidirectional Self-Normalizing Neural Networks

Sitzman et al., in *Implicit Neural Representations With Periodic Activation Functions* propose 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 prove that if the weights are uniformly distributed in $[-c, c]$ where $c = \sqrt{6/f}$ where f is the fan in, then the pre-activations are always standard normal distributed irrespective of the depth of the network. Most importantly, the derivatives of these networks are networks of the same type, ensuring that these guarantees are applicable also to derivatives of networks of this type.

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

Lu et al., in *Bidirectional Self-Normalizing Neural Networks* prove a similar result: 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 $E[f(Z)^2] = 1$ and $E[f'(Z)^2] = 1$ where Z is the normal distribution, and the input vector is thin-shell concentrated in the sense that for all $\epsilon > 0$ $\mathbb{P}\left\{|\frac{1}{n}\|x\|_2^2 - 1| > \epsilon\right\} \to 0$ as $n \to \infty$ then the squared magnitude of the input to each layer is n and the derivative of any loss function E with respect to the input to any layer is the same provided that the layers are wide.

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 the pre-activations are standard normal distributed, but that the derivative of a loss function with respect to the input to any layer always has the same magnitude is an additional guarantee, which when taken together with the first becomes a much stronger assurance with regard to the trainability of deep BSNNs.

Seeing as deeper networks generally have more representational power it is reasonable to hope that such deeper networks could be trained faster and have fewer parameters.

In order to provide these guarantees also for derivatives of the network it is necessary to ensure that $E[f^{(n)}(Z)^2] = 1$ for all n. One solution is $x \mapsto \sqrt{2}\sin(\frac{\pi}{4} + x)$.

To verify that it's a solution, let $f(x) = \sqrt{2}\sin(\frac{\pi}{4} + x)$, then $f'' = -f$, so we can see that it is sufficient to check that $\mathbb{E}[f(Z)^2] = E[f'(Z)^2] = 1$. Now, note that $f(x) = \sin(x) + \cos(x)$. For the first equality $\mathbb{E}[(\sin(Z) + \cos(Z))^2] =$ $= E[\sin^2(Z) + 2\sin(Z)\cos(Z) + \cos(Z)^2] = \mathbb{E}[1 + \sin(2Z)] = 1$. For second equality $\mathbb{E}[(\cos(Z) - \sin(Z))^2] = \mathbb{E}[\cos(Z)^2 - 2\sin(Z)\cos(Z) + \sin^2(Z)] = \mathbb{E}[1 - \sin(2Z)] = 1$.

We now have a concrete class of neural networks that can be trained: networks with linear layers of orthogonal linear transformations followed by this special sine activation function and, ideally, taking input vectors that have squared magnitude equal to their dimension. There are some other solutions for the activation function, for example, $x \mapsto e^{x-1}$,

Positional Encoders

The methods from before are sometimes used to fit neural networks to images or signed distance functions, but the method described above should ideally not be given unprocessed co-ordinates, but something with squared magnitude equal to its dimension.

Mildenhall et al. in their paper *NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis* 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. This can be scaled to produced such vectors, as these vectors always have squared magnitude equal to half their dimension.

This method was originally applied to five-dimensional input, but when applied to two-dimensional input the patterns it introduces in the form of correlations between pixels along lines where one co-ordinate is constant.

These patterns can be removed by applying the positional encoder of Mildenhall et al. to to three new co-ordinate pairs constructed, by, in addition to the original co-ordinate pair constructing two more by rotating the original by one third of a turn and by two thirds of a turn, leading to an improvement in the MSE between the fitted network and the image.

Application to Fitting of Images

Networks of this type, when used with the improved positional encoder can bring MSE between the image and the network output to below 10^{-6} in around 100 iterations and to below 10^{-8} in around 400 iterations, compared to the $10^{-}5$ achieved by Sitzman et al. in their paper, using around 10000 iterations and 90 minutes on reasonable powerful GPU.