
Neural Arithmetic Units

Andreas Madsen^{†‡}
amwebdk@gmail.com

Alexander Rosenberg Johansen[†]
aler@dtu.dk

Who else?

[†]Technical University of Denmark [‡]Computationally Demanding

Abstract

Arithmetic presents unique challenges to machine learning models as they are solved by inferring rules and axioms as opposed to interpolation of data. Neural networks has, with millions and sometimes billions of parameters, an ability to internalize incredible complex functions. However, when extrapolating to out-of-distribution examples they often fail as the neural network learns an approximation and not the underlying logic. We propose a plug-and-play differentiable neural unit that can be trained using stochastic gradient descent and can learn addition, subtraction and multiplication. Our proposed Neural Addition Unit (NAU) and Neural Multiplication Unit (NMU) rely on weight constraints to learn rules and extrapolate well beyond the training distribution. The proposed NAU and NMU are inspired by the Neural Arithmetic Logic Unit (NALU). We find that replacing the nonlinearities in the weight matrix with a clipped linear function and fundamentally reformulate the multiplication unit is crucial for converging consistently. Through analytic and empirical analysis we justify how the NAU and NMU improve the Neural Arithmetic Logic Unit (NALU) and standard multi-layer perceptron (MLP) models. Our models have fewer parameters, convergence more consistently, learns faster and have more meaningful discrete values than the NALU.¹

1 Introduction

The ability for neurons to hold numbers and do arithmetic operations have been documented in both humans, non-human primates ?, as well as newborn chicks ?, and even bees ?. In our race to solve intelligence we have put much faith in neural networks, which in turn has provided unparalleled and often superhuman performance in many tasks requiring high cognitive ability ????. However, when using neural networks to solve simple arithmetic problems, such as counting, they systematically fail to extrapolate ???.

In this paper, we analyze and dramatically improve parts of the recently proposed Neural Arithmetic Logic Unit (NALU) ?. The NALU is a neural network layer with two modules. The two modules, NAC_+ for addition/subtraction and NAC_\bullet for multiplication/division, are selected softly between using a using a sigmoid gating mechanism. By using trainable weights, and restricting the weights

¹In the interest of scientific integrity, we have made the code for all experiments, and more, available on GitHub: <https://github.com/AndreasMadsen/stable-nalu>.

towards $\{-1, 0, 1\}$ the NAC_+ is able to approximate addition and subtraction between hidden units in the previous layer. The NAC_\bullet extends this capability with to achieve multiplication and division. These rules should be learned only by observing arithmetic input-output pairs and using backpropagation?

In practice, training the NALU can be cumbersome. Through an gradient analysis of main components in the NALU, the weight matrix constraint, the multiplication construction, and the gating we present the following findings:

- The weight matrix constraint in the NALU, under zero expectation of the mean layer value?, has a gradient of zero.
- The NAC_\bullet have a treacherous optimization space with unwanted global minimas (as shown in figure 1), exploding/vanishing gradients.
- When performing using the addition module NAC_+ , we observe that the wanted weight matrix values of $-1, 0, 1$ is rarely found.
- The gating between the addition/subtraction component and the multiplication/division component has a gradient that, in expectation, bias' towards the NAC_+ .

Motivated by these convergence and sparsity issue, we propose alternative formulations of the NAC_+ and NAC_\bullet , which we call the Neural Addition Unit (NAU) and Neural Multiplication Unit (NMU). That is, we will assume that the appropriate operation is already known, or can empirically be found by varying the network architecture (oracle gating).

We propose an alternative formulation of the matrix constraint that uses a clipped linear activation, a regularizer that bias towards sparse solutions, and fundamentally reformulate the multiplication unit to be partially linear. All of which significantly improves upon the existing NAC_+ and NAC_\bullet units as shown through extensive testing on synthetic tasks and images.

2 Improving NAC and NALU

The NALU from ? is a neural unit capable of doing either exact addition or multiplication, controlled by a sigmoid-gating-mechanism. The addition part is trivial, as this is just a matrix multiplication $\mathbf{a} = \mathbf{W}\mathbf{x}$. The only special part is that the weight matrix \mathbf{W} is constrained to be between -1 and 1 . This this done using a $\mathbf{W} = \text{tahn}(\hat{\mathbf{W}})\sigma(\hat{\mathbf{M}})$ construction. Meaning that the weight matrix \mathbf{W} is not trained directly, but computed from two auxiliary weight matrices. The core idea is that $\hat{\mathbf{W}}$ controls the sign and $\hat{\mathbf{M}}$ controls if the weight is zero. One of their core claims, is that this weight matrix construction have a sparse bias, which improves extrapolation for cases where a sparse weight is part of the underlying model.

For the multiplication, an exponential-log transformation is used in order to do exact multiplication (within ϵ precision) using a matrix multiplication, $\mathbf{m} = \exp(\mathbf{W} \log(|\mathbf{x}| + \epsilon))$.

The addition unit (originally named NAC), and the multiplication unit are in themselves theoretically applicable in any neural network as well as being differentiable. The NALU, then combines them using a sigmoid-gating-mechanism² $\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$ that chooses softly between addition and multiplication $\mathbf{z} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$.

In terms of the theory, the Original NALU paper ? does not discuss anything more than mentioned so-far in this paper. To aid discussion of why this particular construction problematic, and also

The text before is repeating a lot. Consider just deleting it.

We don't discuss gating

Properly we shouldn't discuss gating in details. And that analysis is extremely complex.

We some wage mention of why we don't consider the gating mechanism.

²The lack of bias term is not a typo. Our preliminary investigations suggests that this is a hack to increase extrapolation of the gate. However in this paper the focus is only arithmetic operators themselves.

suggests improvements which will be empirically validated later, the NAC and its multiplication variant is re-formulated using scalar notation.

$$\begin{aligned}
W_{h_\ell, h_{\ell-1}} &= \tanh(\hat{W}_{h_\ell, h_{\ell-1}}) \sigma(\hat{M}_{h_\ell, h_{\ell-1}}) \\
\text{NAC}_+ : z_{h_\ell} &= \sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} z_{h_{\ell-1}} \\
\text{NAC}_\bullet : z_{h_\ell} &= \exp \left(\sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon) \right)
\end{aligned} \tag{1}$$

2.1 Weight matrix construction

The weight matrix constructions $\tanh(\hat{\mathbf{W}})\sigma(\hat{\mathbf{M}})$ have a few issues worth mentioning. First, the loss gradient with respect to the weight matrices, can without loss of generality, easily be derived to:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{W}_{h_{\ell-1}, h_\ell}} &= \frac{\partial \mathcal{L}}{\partial W_{h_{\ell-1}, h_\ell}} (1 - \tanh^2(\hat{W}_{h_{\ell-1}, h_\ell})) \sigma(\hat{M}_{h_{\ell-1}, h_\ell}) \\
\frac{\partial \mathcal{L}}{\partial \hat{M}_{h_{\ell-1}, h_\ell}} &= \frac{\partial \mathcal{L}}{\partial W_{h_{\ell-1}, h_\ell}} \tanh(\hat{W}_{h_{\ell-1}, h_\ell}) \sigma(\hat{M}_{h_{\ell-1}, h_\ell}) (1 - \sigma(\hat{M}_{h_{\ell-1}, h_\ell}))
\end{aligned} \tag{2}$$

This reveals that this construction is particularly problematic, as $E[\tanh(\hat{W}_{h_{\ell-1}, h_\ell})] = 0$ when $E[\hat{W}_{h_{\ell-1}, h_\ell}] = 0$. Initializing $\hat{W}_{h_{\ell-1}, h_\ell}$ to have zero expectation, is not just common choice but necessary in order to achieve $E[W_{h_{\ell-1}, h_\ell}] = 0$, which is necessary to get desired property $E[z_{h_\ell}] = 0$ in linear units such as as the NAC ?.

The NALU ? paper also claims that this weight matrix construction, creates a bias for $\{-1, 0, 1\}$. However, they provide no empirically or theoretical evidence to support that. In our own empirical investigation as seen in the experiments section, we also find no support for that claim.

To improve on both of these failings, we propose a simple clamped linear construction instead, that is regularize to have the desired bias of $\{-1, 0, 1\}$ and have gradient outside of $[-1, 1]$.

$$\begin{aligned}
W_{h_{\ell-1}, h_\ell} &= \min(\max(\hat{W}_{h_{\ell-1}, h_\ell}, -1), 1), \\
\mathcal{R}_{\ell, \text{bias}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \hat{W}_{h_{\ell-1}, h_\ell}^2 (1 - |\hat{W}_{h_{\ell-1}, h_\ell}|)^2 \\
\mathcal{R}_{\ell, \text{oob}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \max(|\hat{W}_{h_{\ell-1}, h_\ell}| - 1, 0)^2 \\
\text{NAU} : z_{h_\ell} &= \sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} z_{h_{\ell-1}}
\end{aligned} \tag{3}$$

Note that while the bias regularizer $\mathcal{R}_{\ell, \text{bias}}$ also regularize $\hat{W}_{h_{\ell-1}, h_\ell}$ to not be outside of $[-1, 1]$, one may choose a small regularization constant for this, or scale it up gradually as done in the experiments later. However, $\mathcal{R}_{\ell, \text{oob}}$ should always be present as it is never desired to have $\hat{W}_{h_{\ell-1}, h_\ell} \notin [-1, 1]$.

2.2 Multiplication unit

The multiplication unit has its own issues. It should be easy to see that when $|z_{h_{\ell-1}}|$ is near zero and when $\hat{W}_{h_{\ell-1},h_{\ell}}$ is near -1 the $z_{h_{\ell}}$ value explodes. However, the issue extends beyond a weight near -1 as is revealed in the gradients, especially the backpropagation term $\frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{h_{\ell},h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial W_{h_{\ell},h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} z_{h_{\ell}} \log(|z_{h_{\ell-1}}| + \epsilon) \\ \frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} &= \sum_{h_{\ell}=1}^{H_{\ell}} \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} = \sum_{h_{\ell}=1}^{H_{\ell}} \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} z_{h_{\ell}} W_{h_{\ell},h_{\ell-1}} \frac{\text{sign}(z_{h_{\ell-1}})}{|z_{h_{\ell-1}}| + \epsilon}\end{aligned}\quad (4)$$

It should be clear from $\frac{\text{sign}(z_{h_{\ell-1}})}{|z_{h_{\ell-1}}| + \epsilon}$ that for $z_{h_{\ell-1}}$ near zero, the backpropagation term will not only explode, but can oscillate between a large positive value and large negative value, which is very problematic in optimization ?. This issue does not only exist for $|z_{h_{\ell-1}}| < \epsilon$, which may have a small probability if $z_{h_{\ell-1}}$ has a wide distribution. But it can also be an issue for values outside of this interval as seen in figure 1.

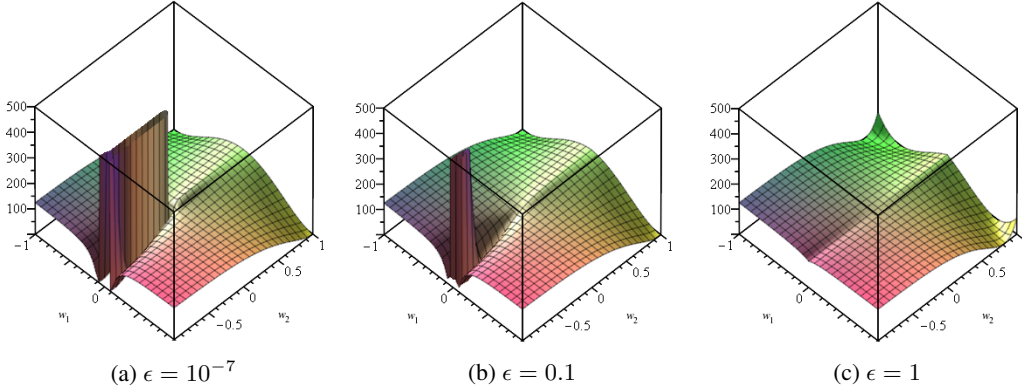


Figure 1: RMS loss curvature for a NAC_+ layer followed by a NAC_\bullet layer. The weight matrices constrained are to $\mathbf{W}_1 = \begin{bmatrix} w_1 & w_1 & 0 & 0 \\ w_1 & w_1 & w_1 & 0 \end{bmatrix}$, $\mathbf{W}_2 = \begin{bmatrix} w_2 & w_2 \end{bmatrix}$. The problem is $x = (1, 1.5, 2, 2)$, $t = 11.25$. Desired solution is $w_1 = w_2 = 1$, although this problem has additional undesired solutions.

These observations are particularly problematic when considering that $E[z_{h_{\ell-1}}] = 0$ is a desired property when initializing ?. An alternative multiplication operator must thus be able to not explode for $z_{h_{\ell-1}}$ near zero. To that end we propose a new neural multiplication unit (NMU):

$$\begin{aligned}W_{h_{\ell-1},h_{\ell}} &= \min(\max(\hat{W}_{h_{\ell-1},h_{\ell}}, 0), 1), \\ \mathcal{R}_{\ell,\text{bias}} &= \frac{1}{H_{\ell} + H_{\ell-1}} \sum_{h_{\ell}=1}^{H_{\ell}} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \hat{W}_{h_{\ell-1},h_{\ell}}^2 (1 - \hat{W}_{h_{\ell-1},h_{\ell}})^2 \\ \mathcal{R}_{\ell,\text{oob}} &= \frac{1}{H_{\ell} + H_{\ell-1}} \sum_{h_{\ell}=1}^{H_{\ell}} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \max\left(\left|\hat{W}_{h_{\ell-1},h_{\ell}} - \frac{1}{2}\right| - \frac{1}{2}, 0\right)^2 \\ \text{NMU} : z_{h_{\ell}} &= \prod_{h_{\ell-1}=1}^{H_{\ell-1}} (W_{h_{\ell-1},h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_{\ell}})\end{aligned}\quad (5)$$

This unit does not support division. But supporting division is likely infeasible if $z_{h_{\ell-1}}$ near zero should not cause explosions. The NALU paper also shows that division doesn't work well for their unit, hence very little is lost here ?. On the other hand, this unit construction understands the

difference between a negative and a positive $z_{h_{\ell-1}}$ values, which should be considered an added bonus, as this allows extrapolations into the negative input range.

The gradients weight gradient and backpropagation term of the NMU are:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{h_{\ell}, h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial W_{h_{\ell}, h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} (z_{h_{\ell-1}} - 1) \\ \frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} &= \sum_{h_{\ell}=1}^{H_{\ell}} \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} = \sum_{h_{\ell}=1}^{H_{\ell}} \frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} W_{h_{\ell-1}, h_{\ell}}\end{aligned}\quad (6)$$

These is much more well-behaved. Note also that the fraction does not explode for $z_{h_{\ell-1}}$ close to zero, as the denominator simply cancels out a term in $z_{h_{\ell}}$.

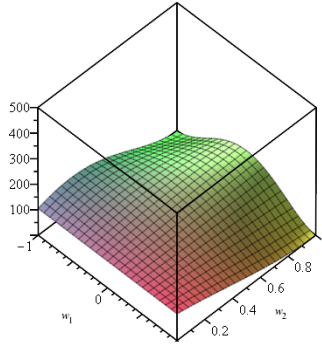


Figure 2: RMS loss curvature (without regularization) for a NAC_+ layer followed by an NMU layer. Otherwise, the setup is identical to that in Figure 1.

2.3 Moments and initialization for addition

Initialization is important for fast and consistent convergence. The desired properties are according to Glorot et al. ?:

$$\begin{aligned}E[z_{h_{\ell}}] &= 0 & E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= 0 \\ \text{Var}[z_{h_{\ell}}] &= \text{Var}[z_{h_{\ell-1}}] & \text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= \text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}}\right]\end{aligned}\quad (7)$$

The NAC_+ layer is trivial, as this is just a linear layer. Thus the result from Glorot et al. ($\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = \frac{2}{H_{\ell-1} + H_{\ell}}$) can be used ?.

In the case of the NAU, this condition is easy to satisfy. However, the original NAC_+ unit is less trivial as $W_{h_{\ell-1}, h_{\ell}}$ is not sampled directly. But assuming that $\hat{W}_{h_{\ell}, h_{\ell-1}} \sim \text{Uniform}[-r, r]$ and $\hat{M}_{h_{\ell}, h_{\ell-1}} \sim \text{Uniform}[-r, r]$ then the variance can be derived to be:

$$\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = \frac{1}{2r} \left(1 - \frac{\tanh(r)}{r}\right) \left(r - \tanh\left(\frac{r}{2}\right)\right) \quad (8)$$

One can the solve for r , given the desired variance.

2.4 Moments and initialization for multiplication

Using second order multivariate Taylor approximation and some assumptions of uncorrelated stochastic variables, the expectation and variance of the NAC_• layer can be estimated to:

$$\begin{aligned}
f(c_1, c_2) &= \left(1 + c_1 \frac{1}{2} \text{Var}[W_{h_\ell, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{c_2 H_{\ell-1}} \\
E[z_{h_\ell}] &\approx f(1, 1) \\
\text{Var}[z_{h_2}] &\approx f(4, 1) - f(1, 2) \\
E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= 0 \\
\text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx \text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell f(4, 1) \text{Var}[W_{h_\ell, h_{\ell-1}}] \\
&\quad \cdot \left(\frac{1}{(|E[z_{h_{\ell-1}}]| + \epsilon)^2} + \frac{3}{(|E[z_{h_{\ell-1}}]| + \epsilon)^4} \text{Var}[z_{h_{\ell-1}}]\right)
\end{aligned} \tag{9}$$

This is problematic because $E[z_{h_\ell}] \geq 1$, and the variance explodes for $E[z_{h_{\ell-1}}] = 0$ which is normally a desired property.

For our proposed NMU, the expectation and variance can be derived using the same assumptions as before, although no Taylor approximation is required:

$$\begin{aligned}
E[z_{h_\ell}] &\approx \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx 0 \\
\text{Var}[z_{h_\ell}] &\approx \left(\text{Var}[W_{h_{\ell-1}, h_\ell}] + \frac{1}{4}\right)^{H_{\ell-1}} (\text{Var}[z_{h_{\ell-1}}] + 1)^{H_{\ell-1}} - \left(\frac{1}{4}\right)^{H_{\ell-1}} \\
\text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx \text{Var}\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell \\
&\quad \cdot \left(\left(\text{Var}[W_{h_{\ell-1}, h_\ell}] + \frac{1}{4}\right)^{H_{\ell-1}} (\text{Var}[z_{h_{\ell-1}}] + 1)^{H_{\ell-1}-1} - \left(\frac{1}{4}\right)^{H_{\ell-1}}\right)
\end{aligned} \tag{10}$$

These expectations are much more well behaved. It is properly unlikely to expect that the expectation can become zero, since the identity for multiplication is 1. However, for a large $H_{\ell-1}$ it will be near zero.

The variance is also more well-behaved, but does not provide a input-independent initialization strategy. We propose initializing with $\text{Var}[W_{h_{\ell-1}, h_\ell}] = \frac{1}{4}$, as this is the solution to $\text{Var}[z_{h_\ell}] = \text{Var}[z_{h_{\ell-1}}]$ assuming $\text{Var}[z_{h_{\ell-1}}] = 1$ and a large $H_{\ell-1}$. However, feel free to compute more exact solutions.

3 Experimental results

3.1 Simple function task

Our simple function task samples an input vector \mathbf{x} from a uniform distribution. From this input vector, the sum of two subsets a and b are then computed. Finally the target t is then an operation performed on a and b (e.g. $a \cdot b$). This is identical to the task by the same name in the Original NALU paper ?. Except that we parameterize it in order to compare the models for different configurations, see figure 3. To make comparison simple, we define a set of default parameters (table 1) and only vary one of them at the time.

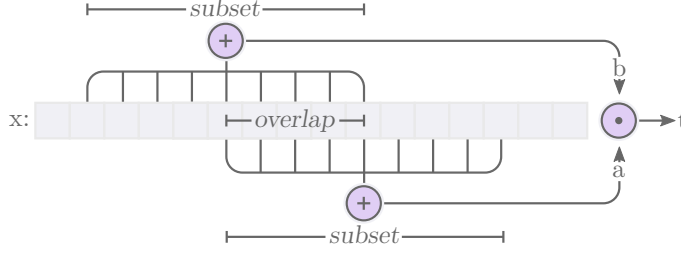


Figure 3: Dataset is parameterized into “Input Size”, “Subset Ratio”, “Overlap Ratio”, an Operation (here showing multiplication), “Interpolation Range” and “Extrapolation Range” from which the data set sampled.

Table 1: Default dataset parameters

Parameter Name	Default Value
Input Size	100
Subset Ratio	0.25
Overlap Ratio	0.5
Interpolation Range	$U[1, 2]$
Extrapolation Range	$U[1, 6]$

Normally one would report the interpolation and extrapolation loss. However, the complex approximations that one would typically see in neural networks are not considered good enough. The goal is to achieve a solution that is sufficiently close to a perfect solution. Because there can be many valid permutations of a perfect solution, especially for addition, a solution is judged first on the final extrapolation error, and then on a sparsity error.

All errors; extrapolation, interpolation, and sparsity are computed every 1000 iterations for 2048 new observations. Because the interpolation and extrapolation errors are quite noisy, even for a near perfect solution. The median over the last 100 measurements is reported.

A model is considered a success if the extrapolation median is less than $\epsilon = 0.2$. This value was acquired by inspecting the error of a near perfect solution.

The sparsity error is computed as in equation 11, and is only considered for the models that did solve the last.

$$E_{\text{sparsity}} = \max_{h_{\ell-1}, h_{\ell}} \min(|W_{h_{\ell-1}, h_{\ell}}|, |1 - |W_{h_{\ell-1}, h_{\ell}}||) \quad (11)$$

The first iteration for which extrapolation $< \epsilon$, is also reported. Again, only models that did solve the task are considered.

For all experiments the $\mathcal{R}_{\ell, \text{oob}}$ regularizer is added to the loss without modification or scaling, while the $\mathcal{R}_{\ell, \text{bias}}$ regularizer is gradually upscaled with $0.1 \cdot (1 - \exp(-10^5 \cdot t))$. Generally this regularizer should just be sufficiently small to not interfere with early training.

3.1.1 Very simple function

To empirically validate the theoretical problems with NAC_•, let’s consider the very simple problem shown earlier in figure 1. That is $x \in \mathbb{R}^4$, $a = x_1 + x_2$ and $b = x_1 + x_2 + x_3 + x_4$. The solution to this problem is that seen in equation 12.

$$\mathbf{W}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{W}_2 = [1 \quad 1] \quad (12)$$

Each model is trained 100 times with different seeds, and stopped after 200000 iterations. Default Adam optimization is used, with a mini-batch size of 128 observations. The results (as seen in table 2), shows that NMU have a much higher success rate and converges much faster. The few cases that did not converge successfully are because of underflow when $w = 0$ in the NMU layer.

Get a better
critation.

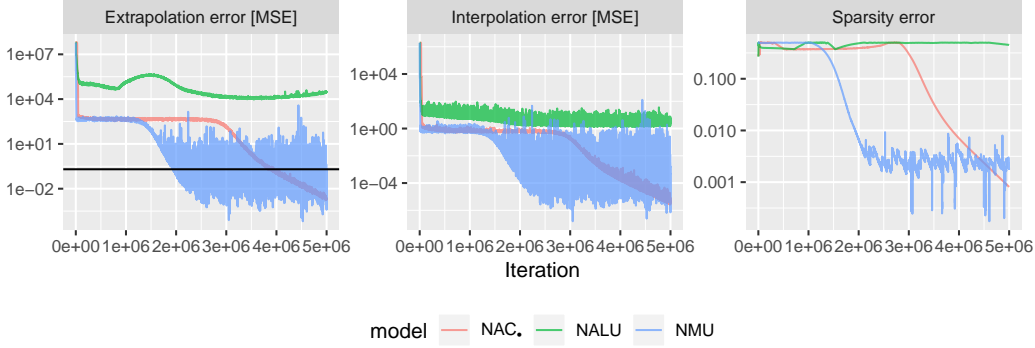


Figure 4: Example of exploration error, interpolation error, and sparsity error, for the task $a \cdot b$ with the default dataset parameters. The horizontal line shows $\epsilon = 0.2$.

Table 2: Shows the success-rate for extrapolation $< \epsilon$, at what global step the model converged at, and the sparsity error for all weight matrices.

Operation	Model	Success rate	Converged at	Sparsity error
$a \cdot b$	NAC \bullet	13%	2969	7.5×10^{-6}
	NALU	26%	3862	9.2×10^{-6}
	NMU	94%	1677	3×10^{-6}

3.1.2 Defaults

To compare on the exact same task as used in the Original NALU paper ?. We report the success rate, the iteration which the model converged, and the sparsity error in table 3. The models are trained for 5000000 iterations. Default Adam optimization is used, with a mini-batch size of 128 observations. The NMU model is an NAU layer followed by an NMU layer. Likewise the NAC \bullet model, is a NAC $+$ layer followed by a NAC \bullet layer.

As seen the NMU model, unlike the NAC \bullet model always converges, and even when NAC \bullet model converges the NMU models converges about twice as fast.

The NAU model, like the NAC $+$ model, always converges. However, NAU model converges more than twice as fast. It even converges faster than a Linear model. Also notice that the NAC $+$ model have a poor sparsity error. This is because it doesn't bias to $\{-1, 0, -1\}$.

Table 3: Shows the success-rate for extrapolation $< \epsilon$, at what global step the model converged at, and the sparsity error for all weight matrices.

Operation	Model	Success rate	Converged at	Sparsity error
$a \cdot b$	NAC \bullet	40%	3371250	4.7×10^{-4}
	NALU	0%	—	—
	NMU	100%	1571900	2.5×10^{-3}
$a - b$	NAC $+$	100%	6300	4.7×10^{-1}
	Linear	100%	3300	3.7×10^{-1}
	NALU	40%	1963250	4.3×10^{-1}
	NAU	100%	3700	1.7×10^{-3}
$a + b$	NAC $+$	100%	42900	4.8×10^{-1}
	Linear	100%	21300	6.1×10^{-1}
	NALU	10%	81000	4.5×10^{-1}
	NAU	100%	15500	2.1×10^{-3}

3.1.3 Exploration of dataset parameters

Finally, the parameters from which the dataset is constructed are considered for just the multiplication problem ($a \cdot b$). The setup is the same the results from table 3. The results are visualized in in figure 5, 6, 8, and 7. Errors bars show the upper and lower 10% quantile, computed over 10 different seeds for each configuration. The center shows the mean of those 10 observations.

Generally the NMU performs far better than both NAC_• and NALU. Some important observations to make:

- Input size > 100 . The NMU model’s success-rate very suddenly decreases when the input size is greater than 100. We have been unable to explain why this happens. We suspect it is a problem with the signal-to-noise ratio of the problem. However the result is also seen if the mini-batch size is dramatically increased.
- Overlap ratio = 0: Both the NMU and also the NAC_• when it does converge, finds a suboptimal solution where in the addition layer $w = 1$ for the overlapping input between a and b , and $w = 0$ for where the input isn’t used. However when an input-scalar is only used in either a or b , convergence the corresponding weights is difficult and slow. Thus the lower the overlap ratio is, the harder the problem is.

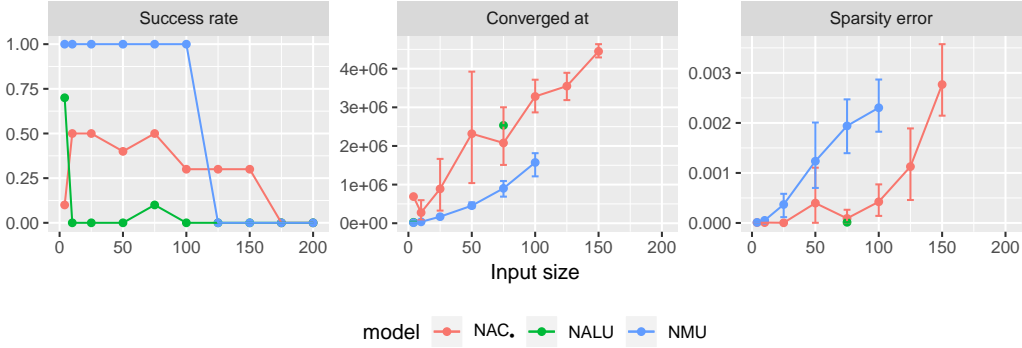


Figure 5: Shows the effect of the input size, on the simple function task problem.

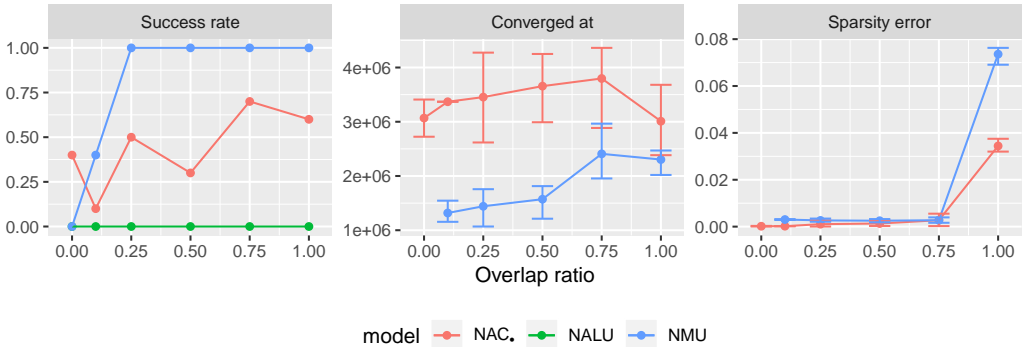


Figure 6: Shows the effect of the overlap ratio, on the simple function task problem.

3.2 Sequential MNIST

To evaluate NALU and NMU in a end-to-end context in combination with a more complex network. We consider the Sequential MNIST Arithmetic task, also presented in the Original NALU paper ?.

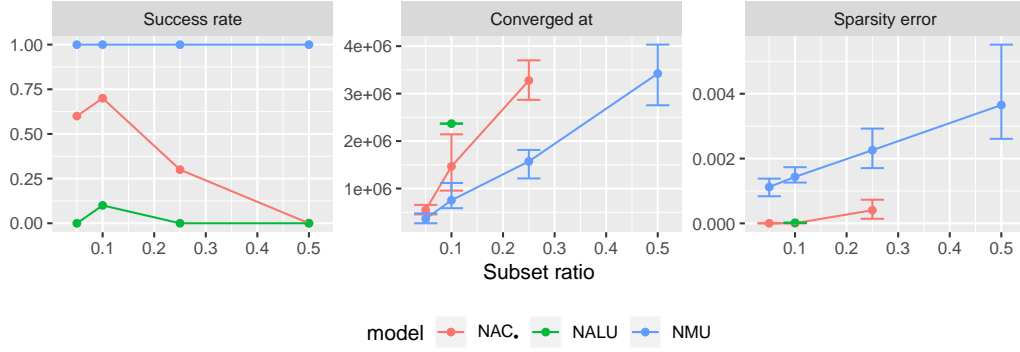


Figure 7: Shows the effect of the subset ratio, on the simple function task problem.

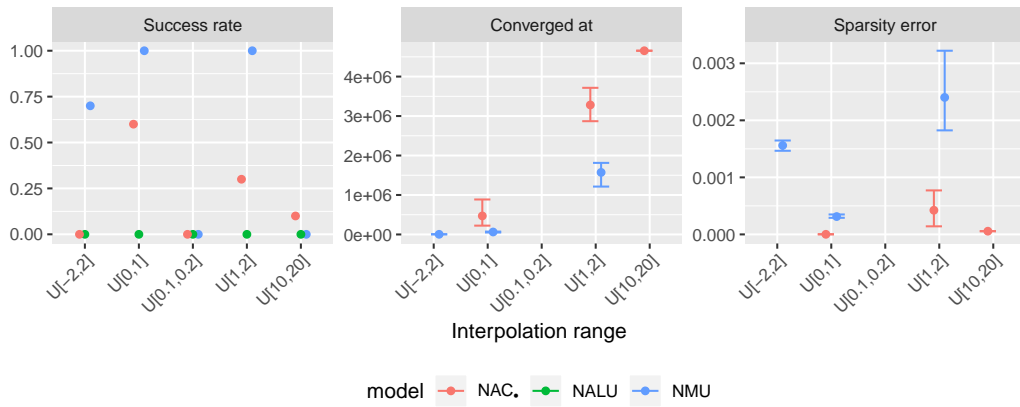


Figure 8: Shows the effect of the interpolation range. For each interpolation range, the following extrapolation ranges are used: $U[-2, 2] \rightarrow U[-6, 6]$, $U[0, 1] \rightarrow U[0, 5]$, $U[0.1, 0.2] \rightarrow U[0, 2]$, $U[1, 2] \rightarrow U[1, 6]$, $U[10, 20] \rightarrow U[1, 40]$.

The task is to take a sequence of MNIST images, then use a CNN layer to produce a hidden layer with 10 elements which somehow describes the number. An recurrent arithmetic unit, is then used to either sum or multiply each MNIST digit together.

This is slightly different from ?, as they only considered addition in the form of counting and a sum. While here consider the sum and the product of a sequence is a considered (figure 9). Such that the multiplication layer also can be judged.

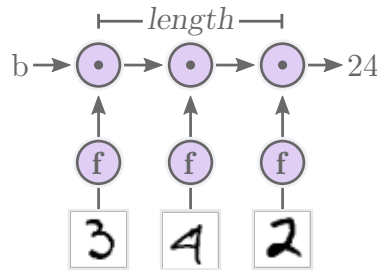


Figure 9: Shows how $3 \cdot 4 \cdot 2$ is computed from a sequence of MNIST digits.

Still waiting for results to be computed.

4 Future work

- Lorem Ipsum

5 Conclusion

Acknowledgments

We would like to thank Andrew Trask and the other authors of the NALU paper, for highlighting the importance and challenges of etrapolation in Neural Networks. We would also like to thank the students Raja Shan Zaker Kreen and William Frisch Moller from The Technical University of Denmark, who showed us that the NALU does not converge consistently.

A Gradient derivatives

A.1 Weight matrix construction

For clarity the weight matrix construction is defined using scalar notation

$$W_{h_\ell, h_{\ell-1}} = \tanh(\hat{W}_{h_\ell, h_{\ell-1}}) \sigma(\hat{M}_{h_\ell, h_{\ell-1}}) \quad (13)$$

The of the loss with respect to $\hat{W}_{h_\ell, h_{\ell-1}}$ and $\hat{M}_{h_\ell, h_{\ell-1}}$ is then straight forward to derive.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \hat{W}_{h_\ell, h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}} \frac{\partial W_{h_\ell, h_{\ell-1}}}{\partial \hat{W}_{h_\ell, h_{\ell-1}}} \\ &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}} (1 - \tanh^2(\hat{W}_{h_\ell, h_{\ell-1}})) \sigma(\hat{M}_{h_\ell, h_{\ell-1}}) \\ \frac{\partial \mathcal{L}}{\partial \hat{M}_{h_\ell, h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}} \frac{\partial W_{h_\ell, h_{\ell-1}}}{\partial \hat{M}_{h_\ell, h_{\ell-1}}} \\ &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}} \tanh(\hat{W}_{h_\ell, h_{\ell-1}}) \sigma(\hat{M}_{h_\ell, h_{\ell-1}}) (1 - \sigma(\hat{M}_{h_\ell, h_{\ell-1}})) \end{aligned} \quad (14)$$

As seen from this result, one only needs to consider $\frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}}$ for NAC_+ and NAC_\bullet , as the gradient with respect to $\hat{W}_{h_\ell, h_{\ell-1}}$ and $\hat{M}_{h_\ell, h_{\ell-1}}$ is just a multiplication on $\frac{\partial \mathcal{L}}{\partial W_{h_\ell, h_{\ell-1}}}$.

A.2 Gradient of NAC_\bullet

First the NAC_\bullet is defined using scalar notation.

$$z_{h_\ell} = \exp \left(\sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon) \right) \quad (15)$$

The gradient of the loss with respect to $W_{h_\ell, h_{\ell-1}}$ is straight forward to derive.

$$\begin{aligned} \frac{\partial z_{h_\ell}}{\partial W_{h_\ell, h_{\ell-1}}} &= \exp \left(\sum_{h'_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h'_{\ell-1}} \log(|z_{h'_{\ell-1}}| + \epsilon) \right) \log(|z_{h_{\ell-1}}| + \epsilon) \\ &= z_{h_\ell} \log(|z_{h_{\ell-1}}| + \epsilon) \end{aligned} \quad (16)$$

We now wish to derive the backpropagation term $\delta_{h_\ell} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}}$, because z_{h_ℓ} affects $\{z_{h_{\ell+1}}\}_{h_{\ell+1}=1}^{H_{\ell+1}}$ this becomes:

$$\delta_{h_\ell} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} = \sum_{h_{\ell+1}=1}^{H_{\ell+1}} \frac{\partial \mathcal{L}}{\partial z_{h_{\ell+1}}} \frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}} = \sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta_{h_{\ell+1}} \frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}} \quad (17)$$

To make it easier to derive $\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}}$ we re-express the z_{h_ℓ} as $z_{h_{\ell+1}}$.

$$z_{h_{\ell+1}} = \exp \left(\sum_{h_\ell=1}^{H_\ell} W_{h_{\ell+1}, h_\ell} \log(|z_{h_\ell}| + \epsilon) \right) \quad (18)$$

The gradient of $\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}}$ is then:

$$\begin{aligned}\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}} &= \exp\left(\sum_{h_{\ell}=1}^{H_{\ell}} W_{h_{\ell+1}, h_{\ell}} \log(|z_{h_{\ell}}| + \epsilon)\right) W_{h_{\ell+1}, h_{\ell}} \frac{\partial \log(|z_{h_{\ell}}| + \epsilon)}{\partial z_{h_{\ell}}} \\ &= \exp\left(\sum_{h_{\ell}=1}^{H_{\ell}} W_{h_{\ell+1}, h_{\ell}} \log(|z_{h_{\ell}}| + \epsilon)\right) W_{h_{\ell+1}, h_{\ell}} \frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \\ &= m_{h_{\ell+1}} W_{h_{\ell+1}, h_{\ell}} \frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon}\end{aligned}\quad (19)$$

$\text{abs}'(z_{h_{\ell}})$ is the gradient of the absolute function. In the paper we denote this as $\text{sign}(z_{h_{\ell}})$ for brevity. However, depending on the exact definition used there may be a difference for $z_{h_{\ell}} = 0$, as $\text{abs}'(0)$ is undefined. In practicality this doesn't matter much though, although theoretically it does mean that the expectation of this is theoretically undefined when $E[z_{h_{\ell}}] = 0$.

A.3 Gradient of NMU

In scalar notation the NMU is defined as:

$$z_{h_{\ell}} = \prod_{h_{\ell-1}=1}^{H_{\ell-1}} (W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}) \quad (20)$$

The gradient of the loss with respect to $W_{h_{\ell-1}, h_{\ell}}$ is fairly trivial. Note that every term but the one for $h_{\ell-1}$, is just a constant with respect to $W_{h_{\ell-1}, h_{\ell}}$. The product, expect the term for $h_{\ell-1}$ can be expressed as $\frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}}$. Using this fact, it becomes trivial to derive the gradient as:

$$\frac{\partial \mathcal{L}}{\partial w_{h_{\ell}, h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial w_{h_{\ell}, h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} (z_{h_{\ell-1}} - 1) \quad (21)$$

Similarly, the gradient $\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}}$ which is essential in backpropagation can equally easily be derived as:

$$\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} = \sum_{h_{\ell}=1}^{H_{\ell}} \frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} = \sum_{h_{\ell}=1}^{H_{\ell}} \frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} W_{h_{\ell-1}, h_{\ell}} \quad (22)$$

B Moments

B.1 Expectation and variance for weight matrix construction in NAC layers

The weight matrix construction in NAC, is defined in scalar notation as:

$$W_{h_{\ell}, h_{\ell-1}} = \tanh(\hat{W}_{h_{\ell}, h_{\ell-1}}) \sigma(\hat{M}_{h_{\ell}, h_{\ell-1}}) \quad (23)$$

Simplifying the notation of this, and re-expressing it using stochastic variables with uniform distributions this can be written as:

$$\begin{aligned}W &\sim \tanh(\hat{W}) \sigma(\hat{M}) \\ \hat{W} &\sim U[-r, r] \\ \hat{M} &\sim U[-r, r]\end{aligned}\quad (24)$$

Since $\tanh(\hat{W})$ is an odd-function and $E[\hat{W}] = 0$, deriving the expectation $E[W]$ is trivial.

$$E[W] = E[\tanh(\hat{W})] E[\sigma(\hat{M})] = 0 \cdot E[\sigma(\hat{M})] = 0 \quad (25)$$

The variance is more complicated, however as \hat{W} and \hat{M} are independent, it can be simplified to:

$$\text{Var}[W] = E[\tanh(\hat{W})^2] E[\sigma(\hat{M})^2] - E[\tanh(\hat{W})]^2 E[\sigma(\hat{M})]^2 = E[\tanh(\hat{W})^2] E[\sigma(\hat{M})^2] \quad (26)$$

These second moments can be analyzed independently. First for $E[\tanh(\hat{W})^2]$:

$$\begin{aligned}
E[\tanh(\hat{W})^2] &= \int_{-\infty}^{\infty} \tanh(x)^2 f_{U[-r,r]}(x) dx \\
&= \frac{1}{2r} \int_{-r}^r \tanh(x)^2 dx \\
&= \frac{1}{2r} \cdot 2 \cdot (r - \tanh(r)) \\
&= 1 - \frac{\tanh(r)}{r}
\end{aligned} \tag{27}$$

Then for $E[\tanh(\hat{M})^2]$:

$$\begin{aligned}
E[\sigma(\hat{M})^2] &= \int_{-\infty}^{\infty} \sigma(x)^2 f_{U[-r,r]}(x) dx \\
&= \frac{1}{2r} \int_{-r}^r \sigma(x)^2 dx \\
&= \frac{1}{2r} \left(r - \tanh\left(\frac{r}{2}\right) \right)
\end{aligned} \tag{28}$$

Finally this gives the variance:

$$\text{Var}[W] = \frac{1}{2r} \left(1 - \frac{\tanh(r)}{r} \right) \left(r - \tanh\left(\frac{r}{2}\right) \right) \tag{29}$$

B.2 Expectation and variance of NAC.

B.2.1 Forward pass

Assuming that each $z_{h_{\ell-1}}$ are independent the expectation can be simplified to:

$$\begin{aligned}
E[z_{h_{\ell}}] &= E \left[\exp \left(\sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_{\ell}, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon) \right) \right] \\
&= E \left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} \exp(W_{h_{\ell}, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon)) \right] \\
&= \prod_{h_{\ell-1}=1}^{H_{\ell-1}} E[\exp(W_{h_{\ell}, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon))] \\
&= E[\exp(W_{h_{\ell}, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon))]^{H_{\ell-1}} \\
&= E \left[(|z_{h_{\ell-1}}| + \epsilon)^{W_{h_{\ell}, h_{\ell-1}}} \right]^{H_{\ell-1}} \\
&= E \left[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}}) \right]^{H_{\ell-1}}
\end{aligned} \tag{30}$$

Here we define f as a non-linear transformation function of two independent stochastic variables:

$$f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}}) = (|z_{h_{\ell-1}}| + \epsilon)^{W_{h_{\ell}, h_{\ell-1}}} \tag{31}$$

We then take the second order taylor approximation of f , around $(E[z_{h_{\ell-1}}], E[W_{h_{\ell}, h_{\ell-1}}])$.

$$\begin{aligned}
E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})] &\approx E \left[\right. \\
&f(E[z_{h_{\ell-1}}], E[W_{h_{\ell}, h_{\ell-1}}]) \\
&+ \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} - E[W_{h_{\ell}, h_{\ell-1}}] \end{bmatrix}^T \begin{bmatrix} \frac{\partial f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial z_{h_{\ell-1}}} \\ \frac{\partial f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial W_{h_{\ell}, h_{\ell-1}}} \end{bmatrix} \left| \begin{array}{l} z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} = E[W_{h_{\ell}, h_{\ell-1}}] \end{array} \right. \\
&+ \frac{1}{2} \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} - E[W_{h_{\ell}, h_{\ell-1}}] \end{bmatrix}^T \\
&\bullet \begin{bmatrix} \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial^2 z_{h_{\ell-1}}} & \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial z_{h_{\ell-1}} \partial W_{h_{\ell}, h_{\ell-1}}} \\ \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial z_{h_{\ell-1}} \partial W_{h_{\ell}, h_{\ell-1}}} & \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial^2 W_{h_{\ell}, h_{\ell-1}}} \end{bmatrix} \left| \begin{array}{l} z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} = E[W_{h_{\ell}, h_{\ell-1}}] \end{array} \right. \\
&\bullet \left. \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} - E[W_{h_{\ell}, h_{\ell-1}}] \end{bmatrix} \right] \quad (32)
\end{aligned}$$

Because $E[z_{h_{\ell-1}} - E[z_{h_{\ell-1}}]] = 0$, $E[W_{h_{\ell}, h_{\ell-1}} - E[W_{h_{\ell}, h_{\ell-1}}]] = 0$, and $Cov[z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}}] = 0$. This simplifies to:

$$\begin{aligned}
E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})] &\approx f(E[z_{h_{\ell-1}}], E[W_{h_{\ell}, h_{\ell-1}}]) \\
&+ \frac{1}{2} Var \begin{bmatrix} z_{h_{\ell-1}} \\ W_{h_{\ell}, h_{\ell-1}} \end{bmatrix}^T \begin{bmatrix} \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial^2 z_{h_{\ell-1}}} \\ \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})}{\partial^2 W_{h_{\ell}, h_{\ell-1}}} \end{bmatrix} \left| \begin{array}{l} z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_{\ell}, h_{\ell-1}} = E[W_{h_{\ell}, h_{\ell-1}}] \end{array} \right. \quad (33)
\end{aligned}$$

Inserting the derivatives and computing the inner products yields:

$$\begin{aligned}
E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})] &\approx (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_{\ell}, h_{\ell-1}}]} \\
&+ \frac{1}{2} Var[z_{h_{\ell-1}}] (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_{\ell}, h_{\ell-1}}]-2} E[W_{h_{\ell}, h_{\ell-1}}] (E[W_{h_{\ell}, h_{\ell-1}}] - 1) \\
&+ \frac{1}{2} Var[W_{h_{\ell}, h_{\ell-1}}] (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_{\ell}, h_{\ell-1}}]} \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \\
&= 1 + \frac{1}{2} Var[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \quad (34)
\end{aligned}$$

This gives the final expectation:

$$\begin{aligned}
E[z_{h_{\ell}}] &= E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})]^{H_{\ell-1}} \\
&\approx \left(1 + \frac{1}{2} Var[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \right)^{H_{\ell-1}} \quad (35)
\end{aligned}$$

As this expectation is of particular interest, we evaluate the error of the approximation, where $W_{h_{\ell}, h_{\ell-1}} \sim U[-r_w, r_w]$ and $z_{h_{\ell-1}} \sim U[0, r_z]$. These distributions are what is used in the simple function task is done. The result is seen in figure 10.

The variance can be derived using the same assumptions about expectation and no correlation.

$$\begin{aligned}
Var[z_{h_{\ell}}] &= E[z_{h_{\ell}}^2] - E[z_{h_{\ell}}]^2 \\
&= E \left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} (|z_{h_{\ell-1}}| + \epsilon)^{2 \cdot W_{h_{\ell}, h_{\ell-1}}} \right] - E \left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} (|z_{h_{\ell-1}}| + \epsilon)^{W_{h_{\ell}, h_{\ell-1}}} \right]^2 \\
&= E[f(z_{h_{\ell-1}}, 2 \cdot W_{h_{\ell}, h_{\ell-1}})]^{H_{\ell-1}} - E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})]^{2 \cdot H_{\ell-1}} \quad (36)
\end{aligned}$$

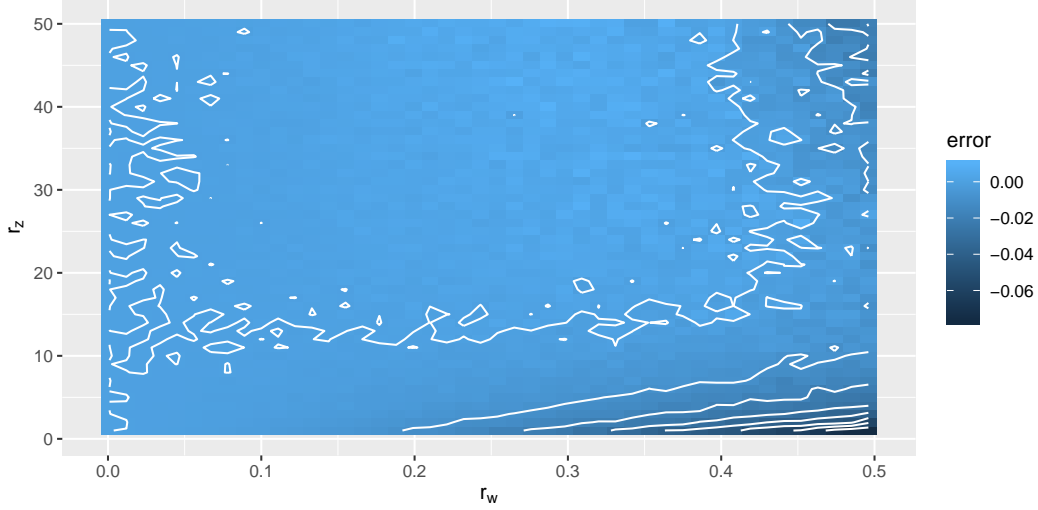


Figure 10: Error between theoretical approximation and the numerical approximation estimated by random sampling of 100000 observations at each combination of r_z and r_w .

We already have from the expectation result that:

$$E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})] \approx 1 + \frac{1}{2} \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \quad (37)$$

By substitution of variable we have that:

$$\begin{aligned} E[f(z_{h_{\ell-1}}, 2 \cdot W_{h_{\ell}, h_{\ell-1}})] &\approx 1 + \frac{1}{2} \text{Var}[2 \cdot W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \\ &\approx 1 + 2 \cdot \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \end{aligned} \quad (38)$$

This gives the variance:

$$\begin{aligned} \text{Var}[z_{h_{\ell}}] &= E[f(z_{h_{\ell-1}}, 2 \cdot W_{h_{\ell}, h_{\ell-1}})]^{H_{\ell-1}} - E[f(z_{h_{\ell-1}}, W_{h_{\ell}, h_{\ell-1}})]^{2 \cdot H_{\ell-1}} \\ &\approx (1 + 2 \cdot \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2)^{H_{\ell-1}} \\ &\quad - \left(1 + \frac{1}{2} \cdot \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{2 \cdot H_{\ell-1}} \end{aligned} \quad (39)$$

B.2.2 Backward pass

The expectation of the backpropagation term:

$$E[\delta_{h_{\ell}}] = E\left[\sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta_{h_{\ell+1}} \frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}}\right] = H_{\ell+1} E[\delta_{h_{\ell+1}}] E\left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}}\right] \quad (40)$$

Where we have that:

$$E\left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}}\right] = E[h_{\ell+1}] E[W_{h_{\ell+1}, h_{\ell}}] E\left[\frac{\text{abs}'(z_{h_{\ell}})}{|z| + \epsilon}\right] = E[m_{h_{\ell+1}}] \cdot 0 \cdot E\left[\frac{\text{abs}'(z_{h_{\ell}})}{|z| + \epsilon}\right] = 0 \quad (41)$$

Deriving the variance is more complicated as:

$$\text{Var}\left[\frac{\partial m_{h_{\ell+1}}}{\partial z_{h_{\ell}}}\right] = \text{Var}\left[m_{h_{\ell+1}} W_{h_{\ell+1}, h_{\ell}} \frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon}\right] \quad (42)$$

Assuming independence between each term this can be simplified to as:

$$\begin{aligned}
\text{Var} \left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_{\ell}}} \right] &= E[z_{h_{\ell+1}}^2] E[W_{h_{\ell+1}, h_{\ell}}^2] E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \right)^2 \right] \\
&\quad - E[z_{h_{\ell+1}}]^2 E[W_{h_{\ell+1}, h_{\ell}}]^2 E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \right)^2 \right] \\
&= E[z_{h_{\ell+1}}^2] \text{Var}[W_{h_{\ell+1}, h_{\ell}}] E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \right)^2 \right] \\
&\quad - E[z_{h_{\ell+1}}]^2 \cdot 0 \cdot E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \right)^2 \right] \\
&= E[z_{h_{\ell+1}}^2] \text{Var}[W_{h_{\ell+1}, h_{\ell}}] E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z_{h_{\ell}}| + \epsilon} \right)^2 \right]
\end{aligned} \tag{43}$$

Using Taylor approximation around $E[z_{h_{\ell}}]$ we have:

$$\begin{aligned}
E \left[\left(\frac{\text{abs}'(z_{h_{\ell}})}{|z| + \epsilon} \right)^2 \right] &\approx \frac{1}{(|E[z_{h_{\ell}}]| + \epsilon)^2} + \frac{1}{2} \frac{6}{(|E[z_{h_{\ell}}]| + \epsilon)^4} \text{Var}[z_{h_{\ell}}] \\
&= \frac{1}{(|E[z_{h_{\ell}}]| + \epsilon)^2} + \frac{3}{(|E[z_{h_{\ell}}]| + \epsilon)^4} \text{Var}[z_{h_{\ell}}]
\end{aligned} \tag{44}$$

Also reusing the result for $E[z_{h_{\ell}}^2]$ from earlier the variance can be expressed as:

$$\begin{aligned}
\text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} \right] &\approx \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} (1 + 2 \cdot \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2)^{H_{\ell-1}} \\
&\quad \cdot \text{Var}[W_{h_{\ell}, h_{\ell-1}}] \left(\frac{1}{(|E[z_{h_{\ell-1}}]| + \epsilon)^2} + \frac{3}{(|E[z_{h_{\ell-1}}]| + \epsilon)^4} \text{Var}[z_{h_{\ell-1}}] \right)
\end{aligned} \tag{45}$$

B.3 Expectation and variance of NMU

B.3.1 Forward pass

Assuming that each $z_{h_{\ell-1}}$ are independent, that $E[z_{h_{\ell-1}}] = 0$, and that $E[W_{h_{\ell-1}, h_{\ell}}] = 1/2$ the expectation is:

$$\begin{aligned}
E[z_{h_{\ell}}] &\approx E \left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} (W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}) \right] \\
&\approx E [W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}]^{H_{\ell-1}} \\
&\approx (E[W_{h_{\ell-1}, h_{\ell}}] E[z_{h_{\ell-1}}] + 1 - E[W_{h_{\ell-1}, h_{\ell}}])^{H_{\ell-1}} \\
&\approx \left(\frac{1}{2} \cdot 0 + 1 - \frac{1}{2} \right)^{H_{\ell-1}} \\
&\approx \left(\frac{1}{2} \right)^{H_{\ell-1}}
\end{aligned} \tag{46}$$

Using the same assumptions for the variance one gets:

$$\begin{aligned}
\text{Var}[z_{h_\ell}] &= E[z_{h_\ell}^2] - E[z_{h_\ell}]^2 \\
&\approx E[z_{h_\ell}^2] - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx E \left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} (W_{h_{\ell-1}, h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_\ell})^2 \right] - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx E[(W_{h_{\ell-1}, h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_\ell})^2]^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx \left(E[W_{h_{\ell-1}, h_\ell}^2] E[z_{h_{\ell-1}}^2] - 2E[W_{h_{\ell-1}, h_\ell}] E[z_{h_{\ell-1}}] \right. \\
&\quad \left. + E[W_{h_{\ell-1}, h_\ell}^2] + 2E[W_{h_{\ell-1}, h_\ell}] E[z_{h_{\ell-1}}] \right. \\
&\quad \left. - 2E[W_{h_{\ell-1}, h_\ell}] + 1 \right)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \tag{47} \\
&\approx \left(E[W_{h_{\ell-1}, h_\ell}^2] E[z_{h_{\ell-1}}^2] + E[W_{h_{\ell-1}, h_\ell}^2] \right. \\
&\quad \left. - 2E[W_{h_{\ell-1}, h_\ell}] + 1 \right)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&= \left(E[W_{h_{\ell-1}, h_\ell}^2] (E[z_{h_{\ell-1}}^2] + 1) \right)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx ((\text{Var}[W_{h_{\ell-1}, h_\ell}] + E[W_{h_{\ell-1}, h_\ell}]^2) (\text{Var}[z_{h_{\ell-1}}] + 1))^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&= \left(\text{Var}[W_{h_{\ell-1}, h_\ell}] + \frac{1}{4} \right)^{H_{\ell-1}} (\text{Var}[z_{h_{\ell-1}}] + 1)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}}
\end{aligned}$$

B.3.2 Backward pass

For the backward pass the expectation can using the same assumptions be derived to:

$$\begin{aligned}
E \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} \right] &= H_\ell E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}} \right] \\
&= H_\ell E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \right] E \left[\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}} \right] \\
&= H_\ell E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \right] E \left[\frac{z_{h_\ell}}{W_{h_{\ell-1}, h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_\ell}} W_{h_{\ell-1}, h_\ell} \right] \\
&= H_\ell E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \right] E \left[\frac{z_{h_\ell}}{W_{h_{\ell-1}, h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_\ell}} \right] E[W_{h_{\ell-1}, h_\ell}] \tag{48} \\
&= H_\ell E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \right] \left(\frac{1}{2}\right)^{H_{\ell-1}-1} \frac{1}{2} \\
&= E \left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \right] H_\ell \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
&\approx 0 \cdot H_\ell \cdot \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
&= 0
\end{aligned}$$

And finally the variance for the backward pass is derived using the same assumptions.

$$\begin{aligned}
\text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} \right] &= H_{\ell} \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} \right] \\
&= H_{\ell} \left(\text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] E \left[\frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} \right]^2 + E \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right]^2 \text{Var} \left[\frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} \right] \right. \\
&\quad \left. + \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] \text{Var} \left[\frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} \right] \right) \\
&\approx \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} \text{Var} \left[\frac{\partial z_{h_{\ell}}}{\partial z_{h_{\ell-1}}} \right] \\
&\approx \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} \left(E \left[\left(\frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} \right)^2 \right] E[W_{h_{\ell-1}, h_{\ell}}^2] \right. \\
&\quad \left. - E \left[\frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} \right]^2 E[W_{h_{\ell-1}, h_{\ell}}]^2 \right) \\
&\approx \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} \left(E \left[\left(\frac{z_{h_{\ell}}}{W_{h_{\ell-1}, h_{\ell}} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1}, h_{\ell}}} \right)^2 \right] E[W_{h_{\ell-1}, h_{\ell}}^2] \right. \\
&\quad \left. - \left(\frac{1}{2} \right)^{2(H_{\ell-1}-1)} \left(\frac{1}{2} \right)^2 \right) \\
&\approx \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} \left(\left(\left(\text{Var}[W_{h_{\ell-1}, h_{\ell}}] + \frac{1}{4} \right) (\text{Var}[z_{h_{\ell-1}}] + 1) \right)^{H_{\ell-1}-1} \right. \\
&\quad \left. \cdot \left(\text{Var}[W_{h_{\ell-1}, h_{\ell}}] + \frac{1}{4} \right) - \left(\frac{1}{2} \right)^{2 \cdot H_{\ell-1}} \right) \\
&= \text{Var} \left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell}}} \right] H_{\ell} \left(\left(\text{Var}[W_{h_{\ell-1}, h_{\ell}}] + \frac{1}{4} \right)^{H_{\ell-1}} (\text{Var}[z_{h_{\ell-1}}] + 1)^{H_{\ell-1}-1} \right. \\
&\quad \left. - \left(\frac{1}{2} \right)^{2 \cdot H_{\ell-1}} \right)
\end{aligned} \tag{49}$$

B.3.3 Initialization

The expectation of $W_{h_{\ell-1}, h_{\ell}}$ should be $E[W_{h_{\ell-1}, h_{\ell}}] = \frac{1}{2}$. Using the variance approximations found, the variance should be according to the forward pass:

$$\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = ((1 + \text{Var}[z_{h_{\ell}}])^{-H_{\ell-1}} \text{Var}[z_{h_{\ell}}] + (4 + 4\text{Var}[z_{h_{\ell}}])^{-H_{\ell-1}})^{\frac{1}{H_{\ell-1}}} - \frac{1}{4} \tag{50}$$

And according to the backward pass it should be:

$$\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = (H_{\ell}(1 + \text{Var}[z_{h_{\ell-1}}])(4 + 4\text{Var}[z_{h_{\ell-1}}])^{-H_{\ell-1}} + (1 + \text{Var}[z_{h_{\ell-1}}])^{1-H_{\ell-1}})^{\frac{1}{H_{\ell-1}}} - \frac{1}{4} \tag{51}$$

These are both dependent on the input variance. If the input variance is known then optimal initialization is possible. However, as this is often not the case one can perhaps assume that $\text{Var}[z_{h_{\ell-1}}] = 1$. This is not an unreasonable assumption in many cases, as there may either be a normalization layer somewhere or the input is normalized. If unit variance is assumed, one gets from the forward pass:

$$\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = (2^{-H_{\ell-1}} + 8^{-H_{\ell-1}})^{\frac{1}{H_{\ell-1}}} - \frac{1}{4} = \frac{1}{8} \left((4^{H_{\ell-1}} + 1)^{H_{\ell-1}} - 2 \right) \tag{52}$$

And from the backward pass:

$$\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = (2H_{\ell}8^{-H_{\ell-1}} + 2^{1-H_{\ell-1}})^{\frac{1}{H}} - \frac{1}{4} \quad (53)$$

The variance requirement for the backward pass is hard to satisfy, as this is dependent on two variables. However, the variance requirement from the forward pass quickly $\text{Var}[W_{h_{\ell-1}, h_{\ell}}] = \frac{1}{4}$ may be a reasonable initialization.

C Simple function task

C.1 Dataset generation

All datasets in the simple function task experiments are generated using the following algorithm:

Algorithm 1 Dataset sampling algorithm

```

1: function DATASET(OP( $\cdot, \cdot$ ) : Operation,  $i$  : InputSize,  $s$  : SubsetRatio,  $o$  : OverlapRatio,
    $R$  : Range)
2:    $\mathbf{x} \leftarrow \text{UNIFORM}(R_{\text{lower}}, R_{\text{upper}}, i)$  ▷ Sample  $i$  elements uniformly
3:    $k \leftarrow \text{UNIFORM}(0, 1 - 2s - o)$  ▷ Sample offset
4:    $a \leftarrow \text{SUM}(\mathbf{x}[ik : i(k + s)])$  ▷ Create sum  $a$  from subset
5:    $b \leftarrow \text{SUM}(\mathbf{x}[i(k + s - o) : i(k + 2s - 0)])$  ▷ Create sum  $b$  from subset
6:    $t \leftarrow \text{OP}(a, b)$  ▷ Perform operation on  $a$  and  $b$ 
7:   return  $x, t$ 

```
