# Neural Arithmetic Units

**Andreas Madsen**[†‡]
amwebdk@gmail.com

**Alexander Rosenberg Johansen**[†]
aler@dtu.dk

**Who else?**

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

## Abstract

Arithmetic problems are solved by rules and axioms, which presents a unique challenge for machine learning models. Neural networks have, with millions and sometimes billions of parameters, an ability to approximate complex functions. However, when extrapolating on out-of-distribution samples neural networks often fail to learn the underlying logic. We propose a plug-and-play differentiable neural unit that can be trained using stochastic gradient descent to 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 has been documented in both humans, non-human primates **?**, newborn chicks **?** and 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 improve parts of the recently proposed Neural Arithmetic Logic Unit (NALU) **?**. Our solution is an alternative formulation of the weight constraint with a clipped linear activation, a regularizer that bias throwards sparse solutions, and a reformulation of the multiplication unit to be partially linear. All of which significantly improves upon the existing $\text{NAC}_+$ and $\text{NAC}_\bullet$ units as shown through extensive testing on synthetic tasks and images.

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

Preprint. Under review.

The NALU is a neural network layer with two sub-units. The two sub-units, $\text{NAC}_+$ for addition/subtraction and $\text{NAC}_\bullet$ for multiplication/division, are softly gated between with a sigmoid function. By using trainable weights, and restricting the weights towards $\{-1, 0, 1\}$. The weights are learned by observing arithmetic input-output pairs and using backpropagation**?**.

We focus only on the $\text{NAC}_+$ and $\text{NAC}_\bullet$ as we have found that the gating in NALU can be cumbersome. This is because of the difficulties in selecting between, and simultaneously training, two vastly different operations.

We will thus assume that the appropriate operation is already known, or can empirically be found by varying the network architecture (oracle gating). We find that the $\text{NAC}_+$ and $\text{NAC}_\bullet$ units poses optimization difficulties. Our analysis presents the following findings:

- The weight matrix constraint in the NALU, under zero expectation of the mean layer value**?**, has a gradient of zero.
- The $\text{NAC}_\bullet$ have a treacherous optimization space with unwanted global minimas (as shown in figure 1) and have exploding/vanishing gradients.
- Using the addition module $\text{NAC}_+$, we observe that the wanted weight matrix values of $\{-1, 0, 1\}$ is rarely found.

Motivated by these convergence and sparsity issue, we propose alternative formulations of the $\text{NAC}_+$ and $\text{NAC}_\bullet$, which we call the Neural Addition Unit (NAU) and Neural Multiplication Unit (NMU).

We have spent a lot of time on gating, and have an intuition on why it might not work. It turns out to be extremely difficult. Unfortunately we have no solution. How do we argue that we don't look at gating?

## 2 Introducing differentiable binary arithmetic operations

Our goal is to achieve arithmetic operations between the elements of a vector. Formally defined as .

$$x_1 \circ_1 x_2 \circ_2 \cdots x_{k-1} \circ_{k-1} x_k \mid (x_1, \ldots, x_k) \in \mathbf{x}, \mathbf{x} \in \mathbb{R}^n, \circ \in \{+, -, \cdot, \div\} \tag{1}$$

Can this be more formally defined

The Neural Arithmetic Logic Unit (NALU)**?** attempts to solve equation 1 by presenting two sub-units; the $\text{NAC}_+$ and $\text{NAC}_\bullet$ to exclusively represent either the $\{+, -\}$ or the $\{\cdot, \div\}$ operations. The NALU attempts to have either $\text{NAC}_+$ or $\text{NAC}_\bullet$ selected exclusively, which could require the NALU to be applied multiple times (alternating between $\text{NAC}_+$ and $\text{NAC}_\bullet$) in order to represent the entire space of solutions for equation 1.

The $\text{NAC}_+$ and $\text{NAC}_\bullet$ are defined accordingly,

$$W_{h_\ell, h_{\ell-1}} = \tanh(\hat{W}_{h_\ell, h_{\ell-1}}) \sigma(\hat{M}_{h_\ell, h_{\ell-1}}) \tag{2}$$

$$\text{NAC}_+ : \; z_{h_\ell} = \sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} z_{h_{\ell-1}} \tag{3}$$

$$\text{NAC}_\bullet : \; z_{h_\ell} = \exp\left( \sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell, h_{\ell-1}} \log(|z_{h_{\ell-1}}| + \epsilon) \right) \tag{4}$$

where $\hat{\mathbf{W}}, \hat{\mathbf{M}} \in \mathbb{R}^{H_\ell \times H_{\ell-1}}$ are trainable weight matrices. These are then combined using tanh and sigmoid transformation to bias the parameters towards a $\{-1, 0, 1\}$ solution. Having $\{-1, 0, 1\}$ allows a linear layer to exactly emulate the binary $\{+, -\}$ operation between elements of a vector as used when computing the $\text{NAC}_+$. The $\text{NAC}_\bullet$ extends the $\text{NAC}_+$ by using an exponential log transformation, which, with $\{-1, 0, 1\}$ weight values, becomes the $\{\cdot, \div\}$ operations (within $\epsilon$ precision).

maybe visualize the function space

The NALU combines these units with a gating mechanism $\mathbf{z} = \mathbf{g} \odot \text{NAC}_+ + (1 - \mathbf{g}) \odot \text{NAC}_\bullet$ given $\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$. The idea is that the NALU should be a plug-and-play component in a neural network and has the ability to, with stochastic gradient descent and backpropagation, to learn the functionality in equation 1.

### 2.1 Challenges of the NALU, NAC$_+$ and NAC$_\bullet$

To simplify the problem we have chosen to leave out the gating mechanism and focus on the sub-units, assuming "oracle gating".

We find this reasonable given ...

We find that gating between NAC$_+$ and NAC$_\bullet$ is challenging to make converge and it it not obvious whether or not the gate could bias towards a specific sub-unit. We have not had any consistent success of convergence using the gating mechanism using the NALU or by combining our own proposed sub-units (NAU, NMU). We find that gating between NAC$_+$ and NAC$_\bullet$ is challenging. This is likely due to the vastly different gradients, causing addition to be learned much faster than multiplication.

In section 2.1.1 we analyse the gradients of equation 2 and propose a cliped linear alternative, with a regularizer that biases towards $\{-1, 0, 1\}$. This results in an alternative addition unit called NAU.

In section 2.1.2 we analyse the gradient of NAC$_\bullet$, and propose an alternative construction called NMU, that has a more well-behaved loss space.

In section 2.1.3 we analyse the moments of NAC$_+$ and NAU, and derive the optimal weight initializations.

In section 2.1.4 we analyse the moments of NAC$_\bullet$ and NMU, and show that the NMU have a more well-behaved expectation and variance.

Maybe remove this overview?

#### 2.1.1 Weight matrix construction

The weight matrix construction $\tanh(\hat{W}_{h_{\ell-1},h_\ell})\sigma(\hat{M}_{h_{\ell-1},h_\ell})$ has the following properties that could make convergence challenging using gradient decent.

The loss gradient with respect to the weight matrices can be derived from equation 2.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{W}_{h_{\ell-1},h_\ell}} &= \frac{\partial \mathcal{L}}{\partial W_{h_{\ell-1},h_\ell}}(1 - \tanh^2(\hat{W}_{h_{\ell-1},h_\ell}))\sigma(\hat{M}_{h_{\ell-1},h_\ell}) \\
\frac{\partial \mathcal{L}}{\partial \hat{M}_{h_{\ell-1},h_\ell}} &= \frac{\partial \mathcal{L}}{\partial W_{h_{\ell-1},h_\ell}} \tanh(\hat{W}_{h_{\ell-1},h_\ell})\sigma(\hat{M}_{h_{\ell-1},h_\ell})(1 - \sigma(\hat{M}_{h_{\ell-1},h_\ell}))
\end{aligned} \tag{5}$$

The gradient $E\left[\partial\mathcal{L}/\partial\hat{M}_{h_{\ell-1},h_\ell}\right] = 0$ can be problematic as we prefer zero having a zero mean expectation of our output. Something that can only be ensured with $E[\hat{W}_{h_{\ell-1},h_\ell}] = 0$ **?**.

In our empirical analysis we find that equation 2 does not create the desired bias for $\{-1, 0, 1\}$, as it doesn't converge towards those values.

To create a bias and prevent the gradient challenges of equation 5 we propose a simple clamped linear construction with an out-of-bound regularizer $\mathcal{R}_{\ell,\text{oob}}$ to force $\hat{W}$ to be within $[-1, 1]$ and ensure that the gradient is always present.

$$\begin{aligned}
W_{h_{\ell-1},h_\ell} &= \min(\max(\hat{W}_{h_{\ell-1},h_\ell}, -1), 1), \\
\mathcal{R}_{\ell,\text{bias}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \hat{W}^2_{h_{\ell-1},h_\ell}(1 - |\hat{W}_{h_{\ell-1},h_\ell}|)^2 \\
\mathcal{R}_{\ell,\text{oob}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \max(|\hat{W}_{h_{\ell-1},h_\ell}| - 1, 0)^2 \\
\text{NAU}: \; z_{h_\ell} &= \sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell,h_{\ell-1}} z_{h_{\ell-1}} \\
\mathcal{L} &= \hat{\mathcal{L}} + \lambda_{\text{bias}}\mathcal{R}_{\ell,\text{bias}} + \lambda_{\text{oob}}\mathcal{R}_{\ell,\text{oob}}
\end{aligned} \tag{6}$$

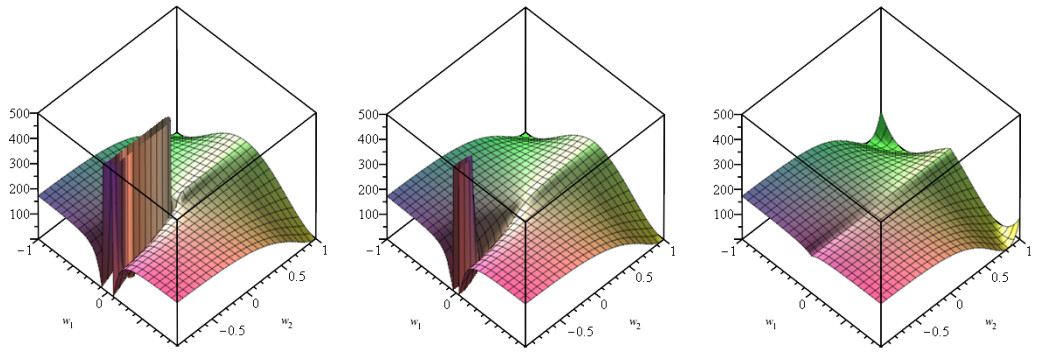
#### 2.1.2 Multiplication unit

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 (see details in Appendix A.2), especially the backpropergation term $\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}$:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial W_{h_\ell,h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} z_{h_\ell} \log(|z_{h_{\ell-1}}| + \epsilon) \\ \frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} &= \sum_{h_\ell=1}^{H_\ell} \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}} = \sum_{h_\ell=1}^{H_\ell} \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} z_{h_\ell} W_{h_\ell,h_{\ell-1}} \frac{\operatorname{sign}(z_{h_{\ell-1}})}{|z_{h_{\ell-1}}| + \epsilon}\end{aligned} \tag{7}$$

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




(a) $\epsilon = 10^{-7}$ (b) $\epsilon = 0.1$ (c) $\epsilon = 1$

Figure 1: RMS loss curvature for a $\mathrm{NAC}_+$ layer followed by a $\mathrm{NAC}_\bullet$ layer. The weight matrices constrained are to $\mathbf{W}_1 = \left[\begin{smallmatrix} w_1 & w_1 & 0 & 0 \\ w_1 & w_1 & w_1 & w_1 \end{smallmatrix}\right]$, $\mathbf{W}_2 = [w_2\ w_2]$. The problem is $x = (1, 1.2, 1.8, 2)\,, t = 13.2$. Desired solution is $w_1 = w_2 = 1$, although this problem have additional undesired solutions.

These observations are particular 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 multplication units (NMU):

$$\begin{aligned} W_{h_{\ell-1},h_\ell} &= \min(\max(\hat{W}_{h_{\ell-1},h_\ell}, 0), 1), \\ \mathcal{R}_{\ell,\text{bias}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \hat{W}^2_{h_{\ell-1},h_\ell} (1 - \hat{W}_{h_{\ell-1},h_\ell})^2 \\ \mathcal{R}_{\ell,\text{oob}} &= \frac{1}{H_\ell + H_{\ell-1}} \sum_{h_\ell=1}^{H_\ell} \sum_{h_{\ell-1}=1}^{H_{\ell-1}} \max\left(\left|\hat{W}_{h_{\ell-1},h_\ell} - \frac{1}{2}\right| - \frac{1}{2}, 0\right)^2 \\ \text{NMU} : z_{h_\ell} &= \prod_{h_{\ell-1}=1}^{H_{\ell-1}} \left(W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right) \end{aligned} \tag{8}$$

This units 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 understand 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 (see details in Appendix A.3):

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial W_{h_\ell,h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}} \left(z_{h_{\ell-1}} - 1\right) \\ \frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} &= \sum_{h_\ell=1}^{H_\ell} \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}} = \sum_{h_\ell=1}^{H_\ell} \frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}} W_{h_{\ell-1},h_\ell}\end{aligned} \tag{9}$$

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 $z_{h_\ell}$.

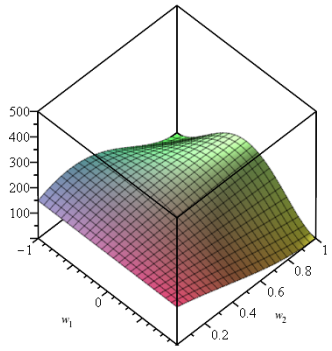



Figure 2: RMS loss curvature (without regularization) for a $\mathrm{NAC}_+$ layer followed by an NMU layer. Otherwise, the setup is identical to that in Figure 1.

### 2.1.3 Moments and initialization for addition

Maybe put this entire section in appendix?

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

$$\begin{aligned} E[z_{h_\ell}] &= 0 & E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= 0 \\ Var[z_{h_\ell}] &= Var\left[z_{h_{\ell-1}}\right] & Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] \end{aligned} \tag{10}$$

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

However, the original $\mathrm{NAC}_+$ unit is less trivial as $W_{h_{\ell-1},h_\ell}$ is not sampled directly. Assuming that $\hat{W}_{h_\ell,h_{\ell-1}} \sim \mathrm{Uniform}[-r, r]$ and $\hat{M}_{h_\ell,h_{\ell-1}} \sim \mathrm{Uniform}[-r, r]$ then the variance can be derived (see proof in Appendix B.1) to be:

$$Var[W_{h_{\ell-1},h_\ell}] = \frac{1}{2r}\left(1 - \frac{\tanh(r)}{r}\right)\left(r - \tanh\left(\frac{r}{2}\right)\right) \tag{11}$$

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

### 2.1.4 Moments and initialization for multiplication

Maybe put this entire section in appendix?

Using second order multivariate Taylor approximation and some assumptionsof uncorrelated

put in assumptions in appendix

stochastic variables, the expectation and variance of the $\mathrm{NAC}_\bullet$ layer can be estimated to:

$$
\begin{aligned}
f(c_1, c_2) &= \left(1 + c_1 \frac{1}{2} Var[W_{h_\ell, h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{c_2 \; H_{\ell-1}} \\
E[z_{h_\ell}] &\approx f\left(1, 1\right) \\
Var[z_{h_2}] &\approx f\left(4, 1\right) - f\left(1, 2\right) \\
E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= 0 \\
Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell \; f\left(4, 1\right) \; Var[W_{h_\ell, h_{\ell-1}}] \\
&\quad \cdot \left(\frac{1}{\left(|E[z_{h_{\ell-1}}]| + \epsilon\right)^2} + \frac{3}{\left(|E[z_{h_{\ell-1}}]| + \epsilon\right)^4} Var[z_{h_{\ell-1}}]\right)
\end{aligned}
\tag{12}
$$

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 (see proof in Appendix B.3) using the same assumptions as before, although no Taylor approximation is required:

$$
\begin{aligned}
E[z_{h_\ell}] &\approx \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx 0 \\
Var[z_{h_\ell}] &\approx \left(Var[W_{h_{\ell-1}, h_\ell}] + \frac{1}{4}\right)^{H_{\ell-1}} \left(Var[z_{h_{\ell-1}}] + 1\right)^{H_{\ell-1}} - \left(\frac{1}{4}\right)^{H_{\ell-1}} \\
Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell \\
&\quad \cdot \left(\left(Var[W_{h_{\ell-1}, h_\ell}] + \frac{1}{4}\right)^{H_{\ell-1}} \left(Var[z_{h_{\ell-1}}] + 1\right)^{H_{\ell-1}-1} - \left(\frac{1}{4}\right)^{H_{\ell-1}}\right)
\end{aligned}
\tag{13}
$$

consider throwing in appendix

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 $Var[W_{h_{\ell-1}, h_\ell}] = \frac{1}{4}$, as this is the solution to $Var[z_{h_\ell}] = Var[z_{h_{\ell-1}}]$ assuming $Var[z_{h_{\ell-1}}] = 1$ and a large $H_{\ell-1}$ (see proof in Appendix B.3.3). However, feel free to compute more exact solutions.

# 3 Experimental results

## 3.1 Simple function task

### 3.1.1 Task definition

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.

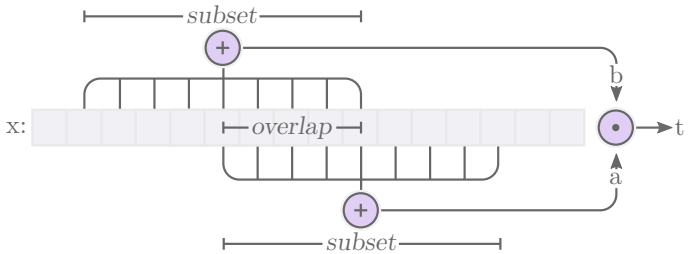



Figure 3: Dataset is parameterized into "Input Size", "Subset Ratio", "Overlap Ratio", an Operation (here showing multiplication), "Interpolation Range" and "Extrapolation Range" from which the data set sampled.

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 | $U[1, 6]$ |

should extrapolation be $U[2, 6]$?

The training set is sampled from the interpolation range throughout training. The extrapolation range is defined as 2048 samples picked for each seed at the start of training.

should we define valid and test sets? maybe we should make one BIG test set and have all models run on that

#### 3.1.2 Criterion

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 firsts on the final extrapolation error, and then on a sparsity error. Because the interpolation and extrapolation errors are quite noisy, even for a near perfect solution. The median over the last 100 measurements is reported.

use the 2048 as a validation set and just pick the best instead. Then report test results for those

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 14, and is only considered for the models that did solve the last.

$$E_{\text{sparsity}} = \max_{h_{\ell-1},h_\ell} \min(|W_{h_{\ell-1},h_\ell}|, |1 - |W_{h_{\ell-1},h_\ell}||) \tag{14}$$

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

Get a better criterion.

#### 3.1.3 Model setup

For all experiements 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 interfer with earily training.

#### 3.1.4 Very simple function

To empirically validate the theoretical problems with $\text{NAC}_\bullet$, 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 15.

$$\mathbf{W}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{W}_2 = [1 \quad 1] \tag{15}$$

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.

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 |
|---|---|---|---|---|
| $a \cdot b$ | $\mathrm{NAC}_\bullet$ | 13% | 29692 | $7.5 \times 10^{-6}$ |
| | NALU | 26% | 38615 | $9.2 \times 10^{-6}$ |
| | NMU | 94% | 16766 | $4.9 \times 10^{-8}$ |

### 3.1.5 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 trainined 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 $\mathrm{NAC}_\bullet$ model, is a $\mathrm{NAC}_+$ layer followed by a $\mathrm{NAC}_\bullet$ layer.

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

The NAU model, like the $\mathrm{NAC}_+$ model, always converges. However, NAU model converges more than twice as fast. It even converges faster than a Linear model. Also notice that the $\mathrm{NAC}_+$ 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 |
|---|---|---|---|---|
| $a \cdot b$ | $\mathrm{NAC}_\bullet$ | 40% | 3371250 | $4.3 \times 10^{-4}$ |
| | Linear | 0% | — | — |
| | NALU | 0% | — | — |
| | NMU | 100% | 1571900 | $1.8 \times 10^{-3}$ |
| $a - b$ | $\mathrm{NAC}_+$ | 100% | 6300 | $4.7 \times 10^{-1}$ |
| | Linear | 100% | 3300 | $3.7 \times 10^{-1}$ |
| | NALU | 40% | 1963250 | $4.3 \times 10^{-1}$ |
| | NAU | 100% | 3700 | $1.8 \times 10^{-3}$ |
| $a + b$ | $\mathrm{NAC}_+$ | 100% | 42900 | $4.8 \times 10^{-1}$ |
| | Linear | 100% | 21300 | $6.1 \times 10^{-1}$ |
| | NALU | 10% | 81000 | $4.5 \times 10^{-1}$ |
| | NAU | 100% | 15500 | $2.2 \times 10^{-3}$ |

add linear to multiplication

### 3.1.6 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 4, 5, 7, and 6. 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 $\mathrm{NAC}_\bullet$ 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-noie 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 $\mathrm{NAC}_\bullet$ 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.

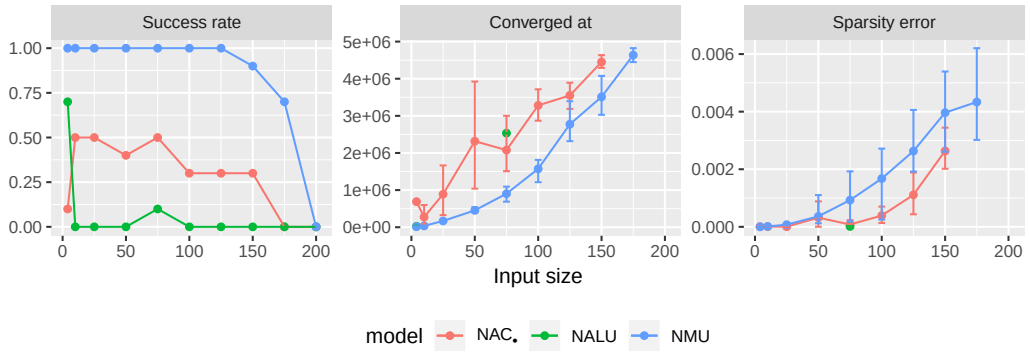



Figure 4: Shows the effect of the input size, on the simple function task problem.

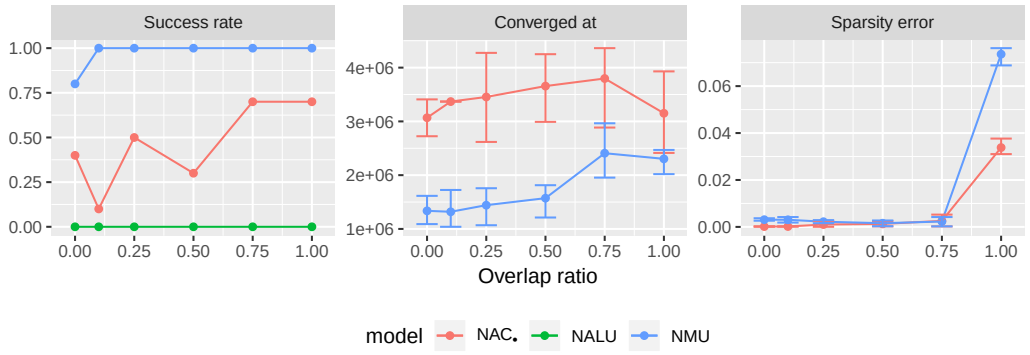



Figure 5: Shows the effect of the overlap ratio, on the simple function task problem.

### 3.2 Sequential MNIST

To evaluate NAU 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 NALU paper **?**.

The task is to take a sequence of MNIST images, then use a CNN layer [2] 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.

[2]We use the model from `https://github.com/pytorch/examples/tree/master/mnist` which is also used in the NALU paper, **?**.

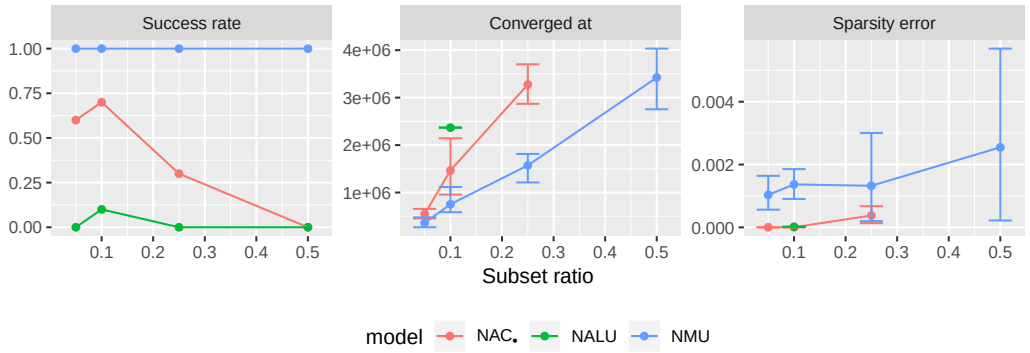



Figure 6: Shows the effect of the subset ratio, on the simple function task problem.

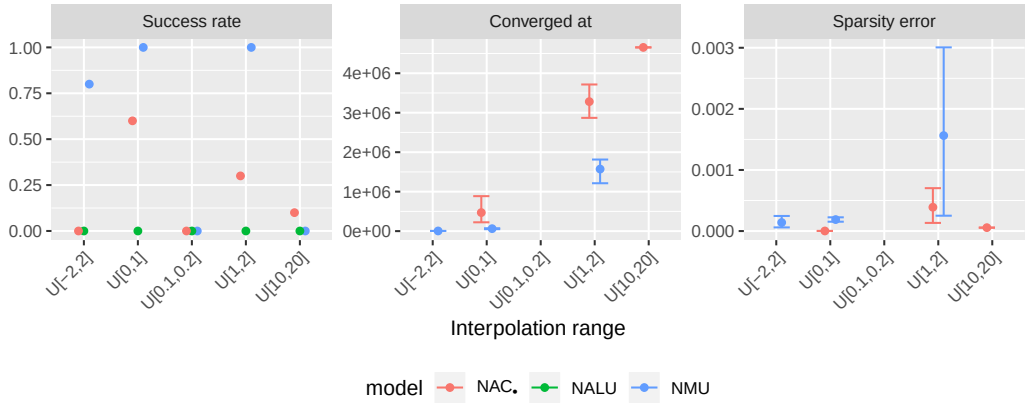



Figure 7: Shows the effect of the interpolation range. For each interpolation range, the following extrapolation ranges are used: $\mathrm{U}[-2,2] \to \mathrm{U}[-6,6]$, $\mathrm{U}[0,1] \to \mathrm{U}[0,5]$, $\mathrm{U}[0.1,0.2] \to \mathrm{U}[0,2]$, $\mathrm{U}[1,2] \to \mathrm{U}[1,6]$, $\mathrm{U}[10,20] \to \mathrm{U}[1,40]$.

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 8). Such that the multiplication layer also can be judged.

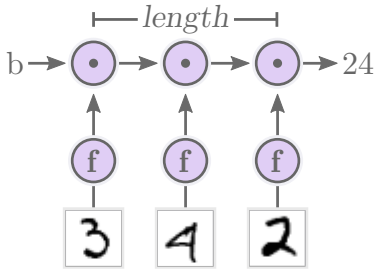



Figure 8: Shows how $3 \cdot 4 \cdot 2$ is computed from a sequence of MNIST digits. $f$ is a CNN layer, that transforms the image into a 10-long vector representation.

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

$$W_{h_\ell,h_{\ell-1}} = \tanh(\hat{W}_{h_\ell,h_{\ell-1}})\sigma(\hat{M}_{h_\ell,h_{\ell-1}}) \tag{16}$$

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.

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \hat{W}_{h_\ell,h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}}\frac{\partial W_{h_\ell,h_{\ell-1}}}{\partial \hat{W}_{h_\ell,h_{\ell-1}}} \\
&= \frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}}(1-\tanh^2(\hat{W}_{h_\ell,h_{\ell-1}}))\sigma(\hat{M}_{h_\ell,h_{\ell-1}}) \\
\frac{\partial \mathcal{L}}{\partial \hat{M}_{h_\ell,h_{\ell-1}}} &= \frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}}\frac{\partial W_{h_\ell,h_{\ell-1}}}{\partial \hat{M}_{h_\ell,h_{\ell-1}}} \\
&= \frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}}\tanh(\hat{W}_{h_\ell,h_{\ell-1}})\sigma(\hat{M}_{h_\ell,h_{\ell-1}})(1-\sigma(\hat{M}_{h_\ell,h_{\ell-1}}))
\end{aligned} \tag{17}$$

As seen from this result, one only needs to consider $\frac{\partial \mathcal{L}}{\partial W_{h_\ell,h_{\ell-1}}}$ for $\mathrm{NAC}_+$ and $\mathrm{NAC}_\bullet$, 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 $\mathrm{NAC}_\bullet$

First the $\mathrm{NAC}_\bullet$ is defined using scalar notation.

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

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

$$\begin{aligned}
\frac{\partial z_{h_\ell}}{\partial W_{h_\ell,h_{\ell-1}}} &= \exp\left(\sum_{h'_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell,h'_{\ell-1}}\log(|z_{h'_{\ell-1}}|+\epsilon)\right)\log(|z_{h_{\ell-1}}|+\epsilon) \\
&= z_{h_\ell}\log(|z_{h_{\ell-1}}|+\epsilon)
\end{aligned} \tag{19}$$

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

$$\delta_{h_\ell} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} = \sum_{h_{\ell+1}=1}^{H_{\ell+1}}\frac{\partial \mathcal{L}}{\partial z_{h_{\ell+1}}}\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}} = \sum_{h_{\ell+1}=1}^{H_{\ell+1}}\delta_{h_{\ell+1}}\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}} \tag{20}$$

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

$$z_{h_{\ell+1}} = \exp\left(\sum_{h_\ell=1}^{H_\ell} W_{h_{\ell+1},h_\ell}\log(|z_{h_\ell}|+\epsilon)\right) \tag{21}$$

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

$$\begin{aligned}\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}} &= \exp\left(\sum_{h_\ell=1}^{H_\ell} W_{h_{\ell+1},h_\ell} \log(|z_{h_\ell}| + \epsilon)\right) W_{h_{\ell+1},h_\ell} \frac{\partial \log(|z_{h_\ell}| + \epsilon)}{\partial z_{h_\ell}} \\ &= \exp\left(\sum_{h_\ell=1}^{H_\ell} W_{h_{\ell+1},h_\ell} \log(|z_{h_\ell}| + \epsilon)\right) W_{h_{\ell+1},h_\ell} \frac{\mathrm{abs}'(z_{h_\ell})}{|z_{h_\ell}| + \epsilon} \\ &= m_{h_{\ell+1}} W_{h_{\ell+1},h_\ell} \frac{\mathrm{abs}'(z_{h_\ell})}{|z_{h_\ell}| + \epsilon}\end{aligned} \tag{22}$$

$\mathrm{abs}'(z_{h_\ell})$ is the gradient of the absolute function. In the paper we denote this as $\mathrm{sign}(z_{h_\ell})$ for brevity. However, depending on the exact defintion used there may be a difference for $z_{h_\ell} = 0$, as $\mathrm{abs}'(0)$ 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:

$$z_{h_\ell} = \prod_{h_{\ell-1}=1}^{H_{\ell-1}} \left(W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right) \tag{23}$$

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:

$$\frac{\partial \mathcal{L}}{\partial w_{h_\ell,h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial w_{h_\ell,h_{\ell-1}}} = \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}} \left(z_{h_{\ell-1}} - 1\right) \tag{24}$$

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

$$\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}} = \sum_{h_\ell=1}^{H_\ell} \frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}} = \sum_{h_\ell=1}^{H_\ell} \frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}} W_{h_{\ell-1},h_\ell} \tag{25}$$

## 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:

$$W_{h_\ell,h_{\ell-1}} = \tanh(\hat{W}_{h_\ell,h_{\ell-1}}) \sigma(\hat{M}_{h_\ell,h_{\ell-1}}) \tag{26}$$

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

$$\begin{aligned} W &\sim \tanh(\hat{W}) \sigma(\hat{M}) \\ \hat{W} &\sim U[-r, r] \\ \hat{M} &\sim U[-r, r] \end{aligned} \tag{27}$$

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

$$\mathrm{E}[W] = \mathrm{E}[\tanh(\hat{W})]\mathrm{E}[\sigma(\hat{M})] = 0 \cdot \mathrm{E}[\sigma(\hat{M})] = 0 \tag{28}$$

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

$$\mathrm{Var}[W] = \mathrm{E}[\tanh(\hat{W})^2]\mathrm{E}[\sigma(\hat{M})^2] - \mathrm{E}[\tanh(\hat{W})]^2\mathrm{E}[\sigma(\hat{M})]^2 = \mathrm{E}[\tanh(\hat{W})^2]\mathrm{E}[\sigma(\hat{M})^2] \tag{29}$$

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

$$\begin{aligned}
\mathrm{E}[\tanh(\hat{W})^2] &= \int_{-\infty}^{\infty} \tanh(x)^2 f_{U[-r,r]}(x)\,\mathrm{d}x \\
&= \frac{1}{2r}\int_{-r}^{r} \tanh(x)^2\,\mathrm{d}x \\
&= \frac{1}{2r}\cdot 2\cdot(r-\tanh(r)) \\
&= 1-\frac{\tanh(r)}{r}
\end{aligned} \tag{30}$$

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

$$\begin{aligned}
\mathrm{E}[\sigma(\hat{M})^2] &= \int_{-\infty}^{\infty} \sigma(x)^2 f_{U[-r,r]}(x)\,\mathrm{d}x \\
&= \frac{1}{2r}\int_{-r}^{r} \sigma(x)^2\,\mathrm{d}x \\
&= \frac{1}{2r}\left(r-\tanh\left(\frac{r}{2}\right)\right)
\end{aligned} \tag{31}$$

Finally this gives the variance:

$$\mathrm{Var}[W] = \frac{1}{2r}\left(1-\frac{\tanh(r)}{r}\right)\left(r-\tanh\left(\frac{r}{2}\right)\right) \tag{32}$$

### B.2 Expectation and variance of $\mathrm{NAC}_\bullet$

#### B.2.1 Forward pass

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

$$\begin{aligned}
E[z_{h_\ell}] &= E\left[\exp\left(\sum_{h_{\ell-1}=1}^{H_{\ell-1}} W_{h_\ell,h_{\ell-1}}\log(|z_{h_{\ell-1}}|+\epsilon)\right)\right] \\
&= E\left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} \exp(W_{h_\ell,h_{\ell-1}}\log(|z_{h_{\ell-1}}|+\epsilon))\right] \\
&= \prod_{h_{\ell-1}=1}^{H_{\ell-1}} E[\exp(W_{h_\ell,h_{\ell-1}}\log(|z_{h_{\ell-1}}|+\epsilon))] \\
&= E[\exp(W_{h_\ell,h_{\ell-1}}\log(|z_{h_{\ell-1}}|+\epsilon))]^{H_{\ell-1}} \\
&= E\left[(|z_{h_{\ell-1}}|+\epsilon)^{W_{h_\ell,h_{\ell-1}}}\right]^{H_{\ell-1}} \\
&= E\left[f(z_{h_{\ell-1}},W_{h_\ell,h_{\ell-1}})\right]^{H_{\ell-1}}
\end{aligned} \tag{33}$$

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

$$f(z_{h_{\ell-1}},W_{h_\ell,h_{\ell-1}}) = (|z_{h_{\ell-1}}|+\epsilon)^{W_{h_\ell,h_{\ell-1}}} \tag{34}$$

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

$$
\begin{aligned}
&E[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})] \approx E\Bigg[ \\
&f(E[z_{h_{\ell-1}}], E[W_{h_\ell,h_{\ell-1}}]) \\
&+ \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} - E[W_{h_\ell,h_{\ell-1}}] \end{bmatrix}^T \begin{bmatrix} \frac{\partial f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial z_{h_{\ell-1}}} \\ \frac{\partial f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial W_{h_\ell,h_{\ell-1}}} \end{bmatrix} \Bigg|_{\substack{z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} = E[W_{h_\ell,h_{\ell-1}}]}} \\
&+ \frac{1}{2} \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} - E[W_{h_\ell,h_{\ell-1}}] \end{bmatrix}^T \\
&\bullet \begin{bmatrix} \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial^2 z_{h_{\ell-1}}} & \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial z_{h_{\ell-1}} \partial W_{h_\ell,h_{\ell-1}}} \\ \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial z_{h_{\ell-1}} \partial W_{h_\ell,h_{\ell-1}}} & \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial^2 W_{h_\ell,h_{\ell-1}}} \end{bmatrix} \Bigg|_{\substack{z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} = E[W_{h_\ell,h_{\ell-1}}]}} \\
&\bullet \begin{bmatrix} z_{h_{\ell-1}} - E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} - E[W_{h_\ell,h_{\ell-1}}] \end{bmatrix} \Bigg]
\end{aligned} \tag{35}
$$

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 similifies to:

$$
\begin{aligned}
&E[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})] \approx f(E[z_{h_{\ell-1}}], E[W_{h_\ell,h_{\ell-1}}]) \\
&+ \frac{1}{2} Var \begin{bmatrix} z_{h_{\ell-1}} \\ W_{h_\ell,h_{\ell-1}} \end{bmatrix}^T \begin{bmatrix} \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial^2 z_{h_{\ell-1}}} \\ \frac{\partial^2 f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})}{\partial^2 W_{h_\ell,h_{\ell-1}}} \end{bmatrix} \Bigg|_{\substack{z_{h_{\ell-1}} = E[z_{h_{\ell-1}}] \\ W_{h_\ell,h_{\ell-1}} = E[W_{h_\ell,h_{\ell-1}}]}}
\end{aligned} \tag{36}
$$

Inserting the derivatives and computing the inner products yields:

$$
\begin{aligned}
&E[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})] \approx (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_\ell,h_{\ell-1}}]} \\
&+ \frac{1}{2} Var[z_{h_{\ell-1}}] (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_\ell,h_{\ell-1}}]-2} E[W_{h_\ell,h_{\ell-1}}](E[W_{h_\ell,h_{\ell-1}}] - 1) \\
&+ \frac{1}{2} Var[W_{h_\ell,h_{\ell-1}}] (|E[z_{h_{\ell-1}}]| + \epsilon)^{E[W_{h_\ell,h_{\ell-1}}]} \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \\
&= 1 + \frac{1}{2} Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2
\end{aligned} \tag{37}
$$

This gives the final expectation:

$$
\begin{aligned}
E[z_{h_\ell}] &= E\left[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})\right]^{H_{\ell-1}} \\
&\approx \left(1 + \frac{1}{2} Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{H_{\ell-1}}
\end{aligned} \tag{38}
$$

As this expectation is of particular interrest, 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 9.

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

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

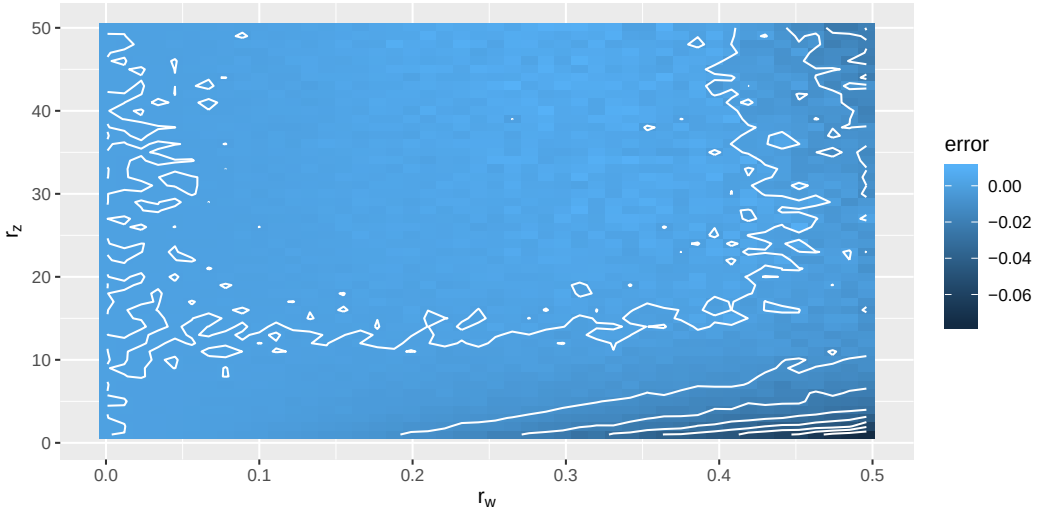



Figure 9: Error between theoretical approximation and the numerical approximation estimated by random sampling of 100000 observations at each combination of $r_z$ and $r_w$.

We already have from the expectation result that:

$$E\left[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})\right] \approx 1 + \frac{1}{2} Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \tag{40}$$

By substitution of variable we have that:

$$\begin{aligned} E\left[f(z_{h_{\ell-1}}, 2 \cdot W_{h_\ell,h_{\ell-1}})\right] &\approx 1 + \frac{1}{2} Var[2 \cdot W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \\ &=\approx 1 + 2 \cdot Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2 \end{aligned} \tag{41}$$

This gives the variance:

$$\begin{aligned} Var[z_{h_\ell}] &= E\left[f(z_{h_{\ell-1}}, 2 \cdot W_{h_\ell,h_{\ell-1}})\right]^{H_{\ell-1}} - E\left[f(z_{h_{\ell-1}}, W_{h_\ell,h_{\ell-1}})\right]^{2 \cdot H_{\ell-1}} \\ &\approx \left(1 + 2 \cdot Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{H_{\ell-1}} \\ &- \left(1 + \frac{1}{2} \cdot Var[W_{h_\ell,h_{\ell-1}}] \log(|E[z_{h_{\ell-1}}]| + \epsilon)^2\right)^{2 \cdot H_{\ell-1}} \end{aligned} \tag{42}$$

### B.2.2 Backward pass

The expectation of the backpropagation term:

$$E[\delta_{h_\ell}] = E\left[\sum_{h_{\ell+1}=1}^{H_{\ell+1}} \delta_{h_{\ell+1}} \frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}}\right] = H_{\ell+1} E[\delta_{h_{\ell+1}}] E\left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}}\right] \tag{43}$$

Where we have that:

$$E\left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}}\right] = E[h_{\ell+1}] E[W_{h_{\ell+1},h_\ell}] E\left[\frac{\text{abs}'(z_{h_\ell})}{|z| + \epsilon}\right] = E[m_{h_{\ell+1}}] \cdot 0 \cdot E\left[\frac{\text{abs}'(z_{h_\ell})}{|z| + \epsilon}\right] = 0 \tag{44}$$

Deriving the variance is more complicated as:

$$Var\left[\frac{\partial m_{h_{\ell+1}}}{\partial z_{h_\ell}}\right] = Var\left[m_{h_{\ell+1}} W_{h_{\ell+1},h_\ell} \frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}| + \epsilon}\right] \tag{45}$$

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

$$\begin{aligned}
Var\left[\frac{\partial z_{h_{\ell+1}}}{\partial z_{h_\ell}}\right] &= E[z_{h_{\ell+1}}^2]E[W_{h_{\ell+1},h_\ell}^2]E\left[\left(\frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}|+\epsilon}\right)^2\right] \\
&\quad - E[z_{h_{\ell+1}}]^2E[W_{h_{\ell+1},h_\ell}]^2E\left[\frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}|+\epsilon}\right]^2 \\
&= E[z_{h_{\ell+1}}^2]Var[W_{h_{\ell+1},h_\ell}]E\left[\left(\frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}|+\epsilon}\right)^2\right] \\
&\quad - E[z_{h_{\ell+1}}]^2 \cdot 0 \cdot E\left[\frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}|+\epsilon}\right]^2 \\
&= E[z_{h_{\ell+1}}^2]Var[W_{h_{\ell+1},h_\ell}]E\left[\left(\frac{\text{abs}'(z_{h_\ell})}{|z_{h_\ell}|+\epsilon}\right)^2\right]
\end{aligned} \tag{46}$$

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

$$\begin{aligned}
E\left[\left(\frac{\text{abs}'(z_{h_\ell})}{|z|+\epsilon}\right)^2\right] &\approx \frac{1}{\left(|E[z_{h_\ell}]|+\epsilon\right)^2} + \frac{1}{2}\frac{6}{\left(|E[z_{h_\ell}]|+\epsilon\right)^4}Var[z_{h_\ell}] \\
&= \frac{1}{\left(|E[z_{h_\ell}]|+\epsilon\right)^2} + \frac{3}{\left(|E[z_{h_\ell}]|+\epsilon\right)^4}Var[z_{h_\ell}]
\end{aligned} \tag{47}$$

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

$$\begin{aligned}
Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] \approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell \ \left(1 + 2 \cdot Var[W_{h_\ell,h_{\ell-1}}]\log(|E[z_{h_{\ell-1}}]|+\epsilon)^2\right)^{H_{\ell-1}} \\
\cdot Var[W_{h_\ell,h_{\ell-1}}]\left(\frac{1}{\left(|E[z_{h_{\ell-1}}]|+\epsilon\right)^2} + \frac{3}{\left(|E[z_{h_{\ell-1}}]|+\epsilon\right)^4}Var[z_{h_{\ell-1}}]\right)
\end{aligned} \tag{48}$$

### 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:

$$\begin{aligned}
E[z_{h_\ell}] &\approx E\left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}}\left(W_{h_{\ell-1},h_\ell}z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right)\right] \\
&\approx E\left[W_{h_{\ell-1},h_\ell}z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right]^{H_{\ell-1}} \\
&\approx \left(E[W_{h_{\ell-1},h_\ell}]E[z_{h_{\ell-1}}] + 1 - E[W_{h_{\ell-1},h_\ell}]\right)^{H_{\ell-1}} \\
&\approx \left(\frac{1}{2}\cdot 0 + 1 - \frac{1}{2}\right)^{H_{\ell-1}} \\
&\approx \left(\frac{1}{2}\right)^{H_{\ell-1}}
\end{aligned} \tag{49}$$

Using the same assumptions for the variance one gets:

$$
\begin{aligned}
Var[z_{h_\ell}] &= E[z_{h_\ell}^2] - E[z_{h_\ell}]^2 \\
&\approx E[z_{h_\ell}^2] - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx E\left[\prod_{h_{\ell-1}=1}^{H_{\ell-1}} \left(W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right)^2\right] - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx E[\left(W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}\right)^2]^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx \Big(E[W_{h_{\ell-1},h_\ell}^2] E[z_{h_{\ell-1}}^2] - 2E[W_{h_{\ell-1},h_\ell}^2] E[z_{h_{\ell-1}}] \\
&\quad + E[W_{h_{\ell-1},h_\ell}^2] + 2E[W_{h_{\ell-1},h_\ell}] E[z_{h_{\ell-1}}] \\
&\quad - 2E[W_{h_{\ell-1},h_\ell}] + 1\Big)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx \Big(E[W_{h_{\ell-1},h_\ell}^2] E[z_{h_{\ell-1}}^2] + E[W_{h_{\ell-1},h_\ell}^2] \\
&\quad - 2E[W_{h_{\ell-1},h_\ell}] + 1\Big)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&= \Big(E[W_{h_{\ell-1},h_\ell}^2] \left(E[z_{h_{\ell-1}}^2] + 1\right)\Big)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&\approx \left(\left(Var[W_{h_{\ell-1},h_\ell}] + E[W_{h_{\ell-1},h_\ell}]^2\right) \left(Var[z_{h_{\ell-1}}] + 1\right)\right)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}} \\
&= \left(Var[W_{h_{\ell-1},h_\ell}] + \frac{1}{4}\right)^{H_{\ell-1}} \left(Var[z_{h_{\ell-1}}] + 1\right)^{H_{\ell-1}} - \left(\frac{1}{2}\right)^{2 \cdot H_{\ell-1}}
\end{aligned}
\tag{50}
$$

### B.3.2 Backward pass

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

$$
\begin{aligned}
E\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= H_\ell E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}} \frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right] \\
&= H_\ell E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] E\left[\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right] \\
&= H_\ell E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] E\left[\frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}} W_{h_{\ell-1},h_\ell}\right] \\
&= H_\ell E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] E\left[\frac{z_{h_\ell}}{W_{h_{\ell-1},h_\ell} z_{h_{\ell-1}} + 1 - W_{h_{\ell-1},h_\ell}}\right] E\left[W_{h_{\ell-1},h_\ell}\right] \\
&= H_\ell E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] \left(\frac{1}{2}\right)^{H_{\ell-1}-1} \frac{1}{2} \\
&= E\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right] H_\ell \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
&\approx 0 \cdot H_\ell \cdot \left(\frac{1}{2}\right)^{H_{\ell-1}} \\
&= 0
\end{aligned}
\tag{51}
$$

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

$$
\begin{aligned}
Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_{\ell-1}}}\right] &= H_\ell Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right] \\
&= H_\ell\Bigg(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 Var\left[\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right] \\
&+ Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]Var\left[\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right]\Bigg) \\
&\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]H_\ell Var\left[\frac{\partial z_{h_\ell}}{\partial z_{h_{\ell-1}}}\right] \\
&\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]H_\ell\Bigg(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] \\
&- 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\Bigg) \\
&\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]H_\ell\Bigg(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] \\
&- \left(\frac{1}{2}\right)^{2\cdot(H_{\ell-1}-1)}\left(\frac{1}{2}\right)^2\Bigg) \\
&\approx Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]H_\ell\Bigg(\left(\left(Var[W_{h_{\ell-1},h_\ell}]+\frac{1}{4}\right)\left(Var[z_{h_{\ell-1}}]+1\right)\right)^{H_{\ell-1}-1} \\
&\cdot\left(Var[W_{h_{\ell-1},h_\ell}]+\frac{1}{4}\right)-\left(\frac{1}{2}\right)^{2\cdot H_{\ell-1}}\Bigg) \\
&= Var\left[\frac{\partial \mathcal{L}}{\partial z_{h_\ell}}\right]H_\ell\Bigg(\left(Var[W_{h_{\ell-1},h_\ell}]+\frac{1}{4}\right)^{H_{\ell-1}}\left(Var[z_{h_{\ell-1}}]+1\right)^{H_{\ell-1}-1} \\
&- \left(\frac{1}{2}\right)^{2\cdot H_{\ell-1}}\Bigg)
\end{aligned}
\tag{52}
$$

### 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:

$$
Var[W_{h_{\ell-1},h_\ell}] = \left((1+Var[z_{h_\ell}])^{-H_{\ell-1}}Var[z_{h_\ell}] + (4+4Var[z_{h_\ell}])^{-H_{\ell-1}}\right)^{\frac{1}{H_{\ell-1}}} - \frac{1}{4} \tag{53}
$$

And according to the backward pass it should be:

$$
Var[W_{h_{\ell-1},h_\ell}] = \left(H_\ell(1+Var[z_{h_{\ell-1}}])(4+4Var[z_{h_{\ell-1}}])^{-H_{\ell-1}} + (1+Var[z_{h_{\ell-1}}])^{1-H_{\ell-1}}\right)^{\frac{1}{H}} - \frac{1}{4} \tag{54}
$$

These are both dependent on the input variance. If the input variance is know then optimal initialization is possible. However, as this is often not the case one can perhaps assume that $Var[z_{h_{\ell-1}}] = 1$. 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:

$$
Var[W_{h_{\ell-1},h_\ell}] = \left(2^{-H_{\ell-1}} + 8^{-H_{\ell-1}}\right)^{\frac{1}{H_{\ell-1}}} - \frac{1}{4} = \frac{1}{8}\left(\left(4^{H_{\ell-1}}+1\right)^{H_{\ell-1}} - 2\right) \tag{55}
$$

And from the backward pass:

$$Var[W_{h_{\ell-1},h_\ell}] = \left(2H_\ell 8^{-H_{\ell-1}} + 2^{1-H_{\ell-1}}\right)^{\frac{1}{H}} - \frac{1}{4} \tag{56}$$

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 $Var[W_{h_{\ell-1},h_\ell}] = \frac{1}{4}$ 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

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