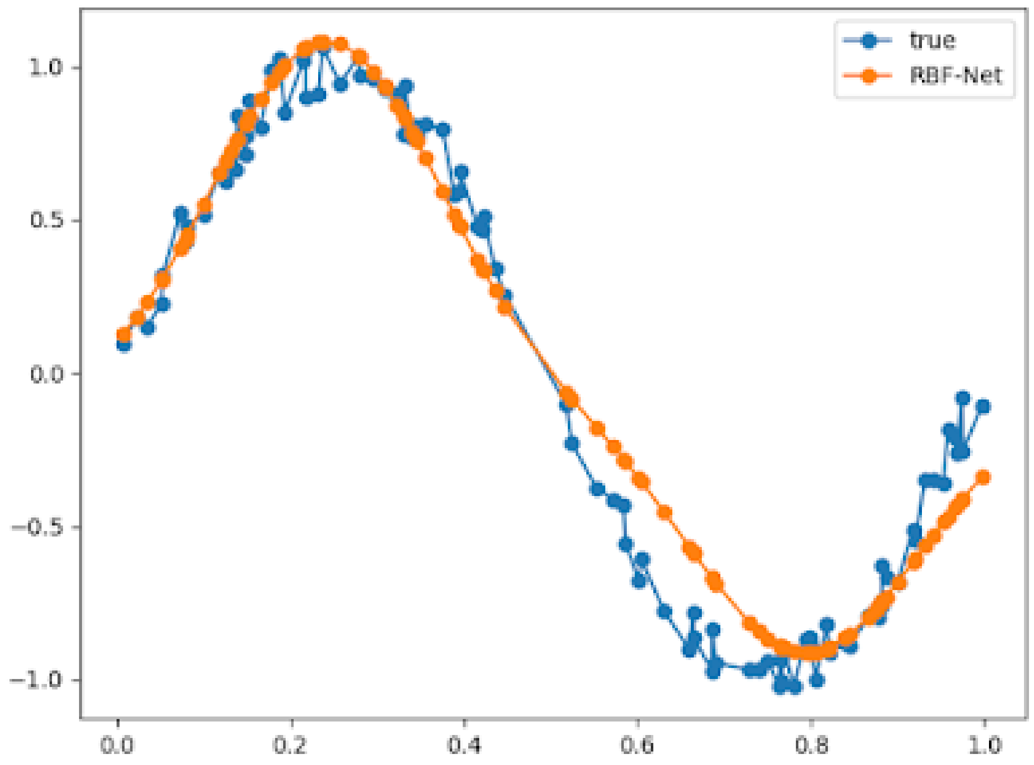
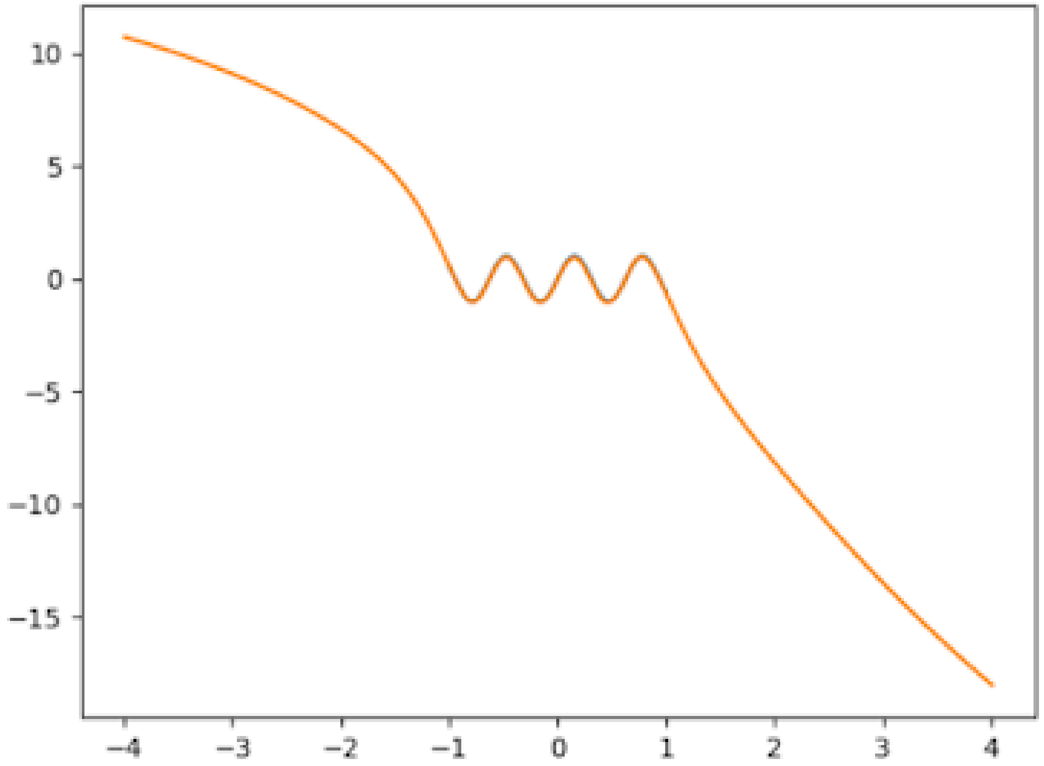
Neural Arithmetic Logic Units

한양대학교 AILAB 석사과정 엄희송


true
RBF-Net
1.0
0.5
0.0
−0.5
−1.0
0.0
0.2
0.4
0.6
0.8
1.0


10
5
0
−5
−10
−15
−4
−3
−2
−1
0
1
2
3
4

각각의 activation function을 사용하는 MLP들이 identity function(Y=X)를 학습하는 과정에서 error의 모습.

Training range 안에서는 낮은 error,
그 범위를 벗어나면 error가 증가함.

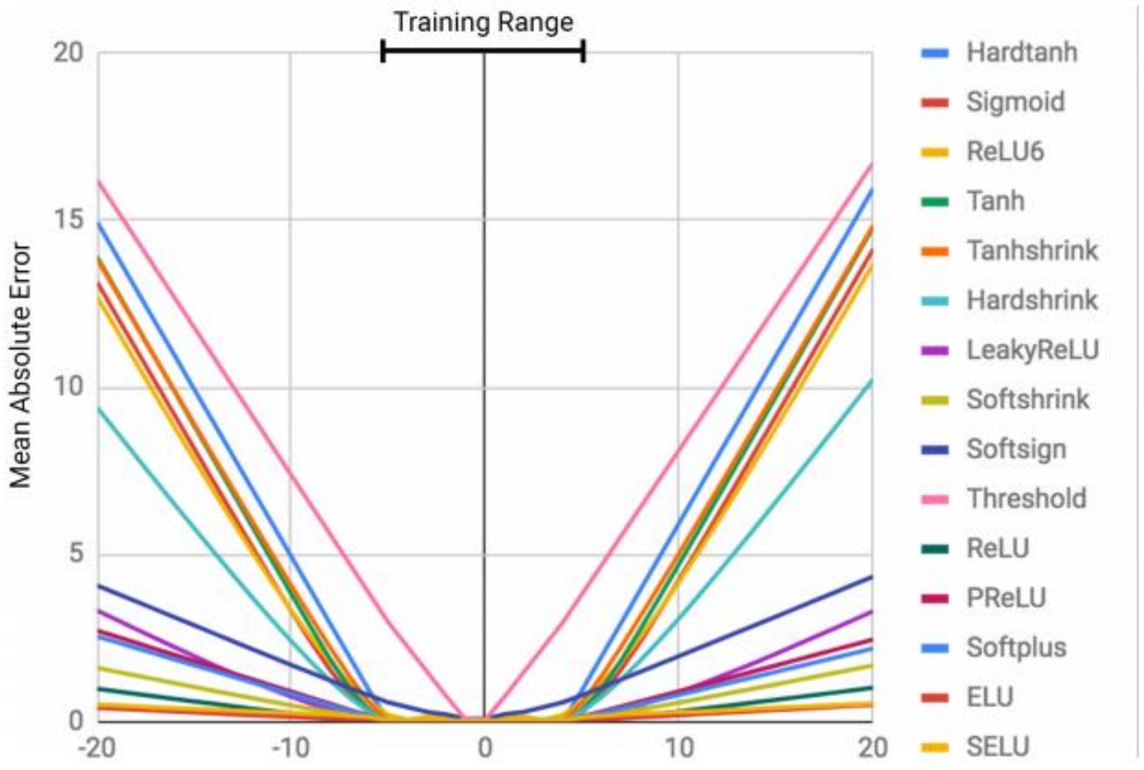
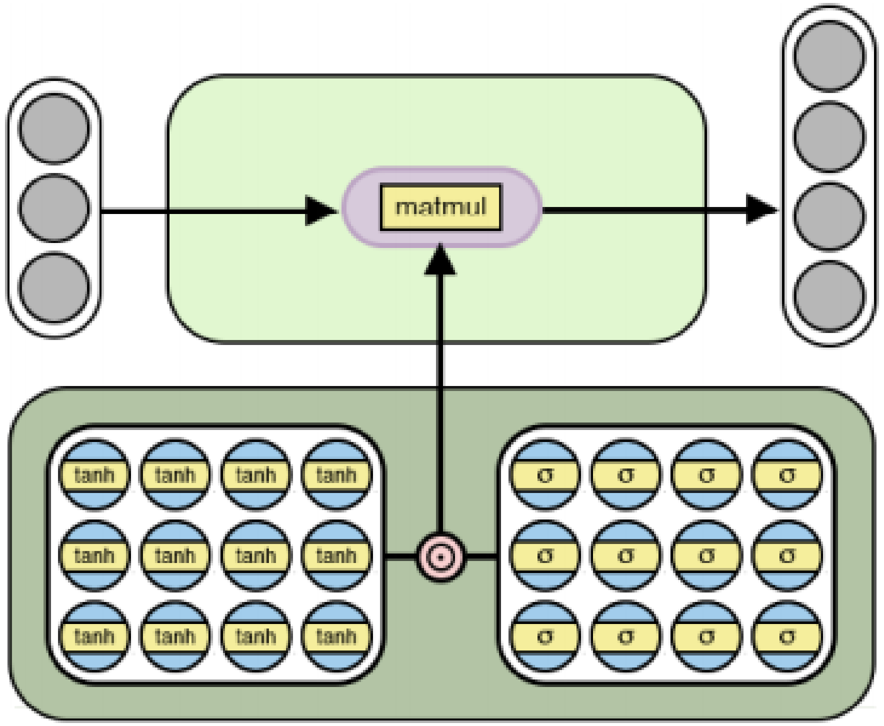
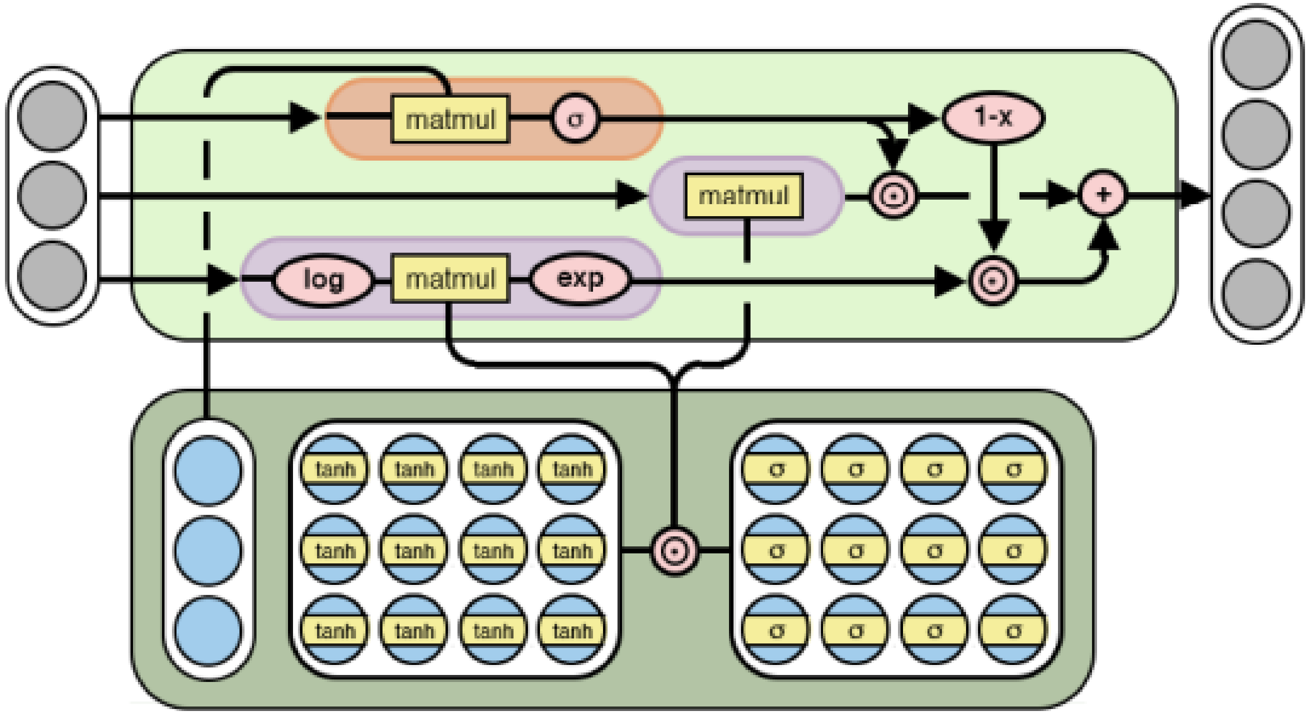


Figure 1: MLPs learn the identity function only for the range of values they are trained on. The mean error ramps up severely both below and above the range of numbers seen during training.

NAC & NALU



(a) Neural Accumulator (NAC)

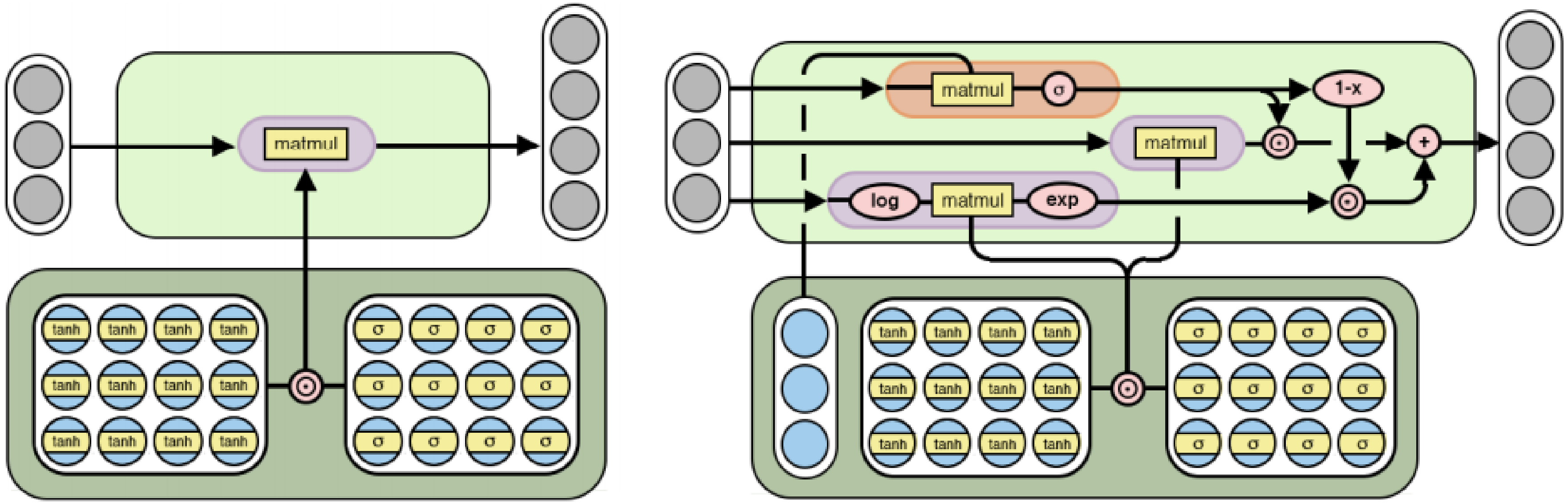


(b) Neural Arithmetic Logic Unit (NALU)

$\widehat{W}, \widehat{M}$, G shape: [input length, output length]

$$\texttt{NAC:} \quad \mathbf{a} = \mathbf{W}\mathbf{x} \qquad \mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m} \qquad \mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon)), \; \mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$$

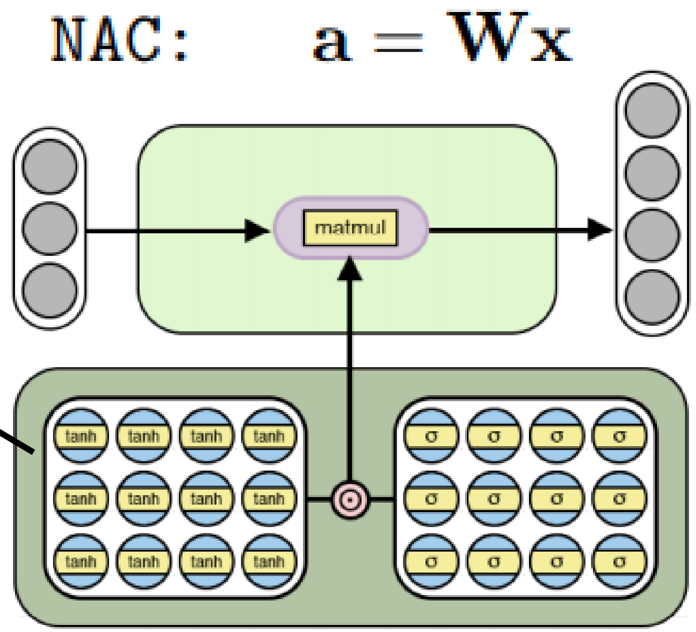


(a) Neural Accumulator (NAC) (b) Neural Arithmetic Logic Unit (NALU)

$$\mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

학습이 완료된 W matrix는
각각의 element가 {-1, 0, 1} 중
하나의 값에 가까워지게 된다.

W를 tanh 와 sigmoid의 element-wise multiplication
하는 이유는 gradient를 통해 학습시키고 싶어서



(a) Neural Accumulator (NAC)

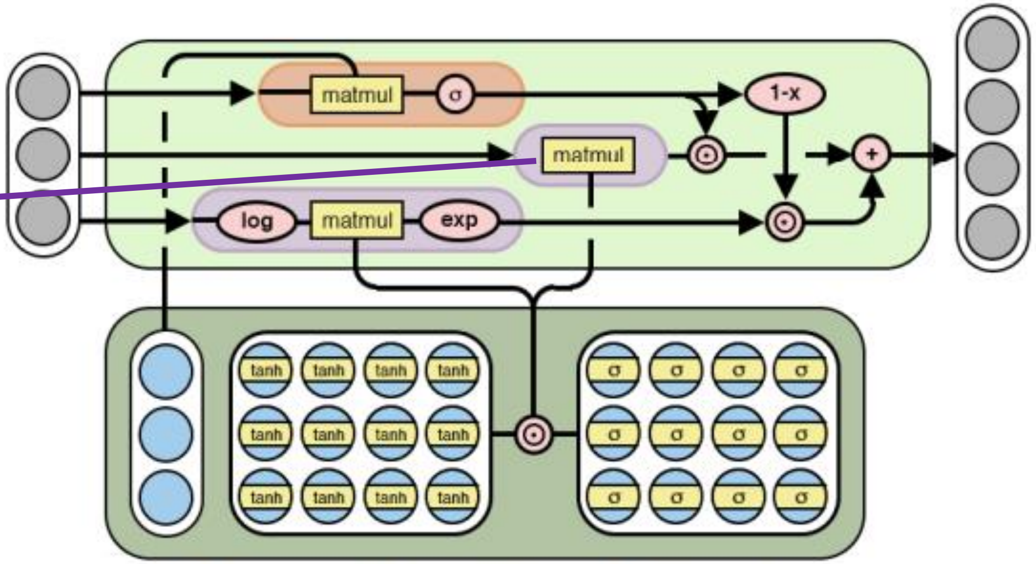
$$\mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

$$\boxed{\texttt{NAC:} \quad \mathbf{a} = \mathbf{W}\mathbf{x}}$$

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

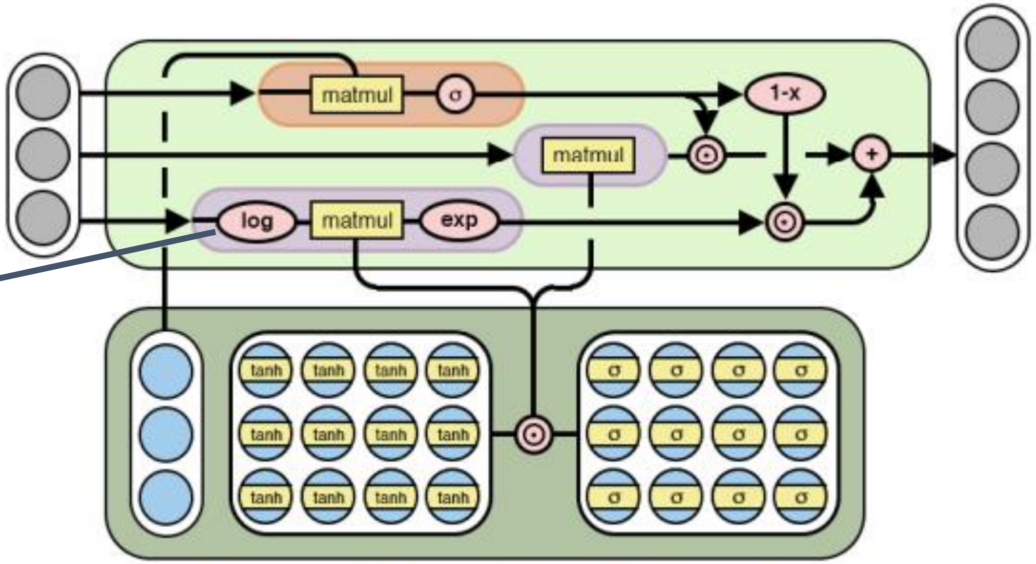
$$\mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

$$\texttt{NAC:} \quad \mathbf{a} = \mathbf{W}\mathbf{x}$$

$$\boxed{\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))}$$

$$\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

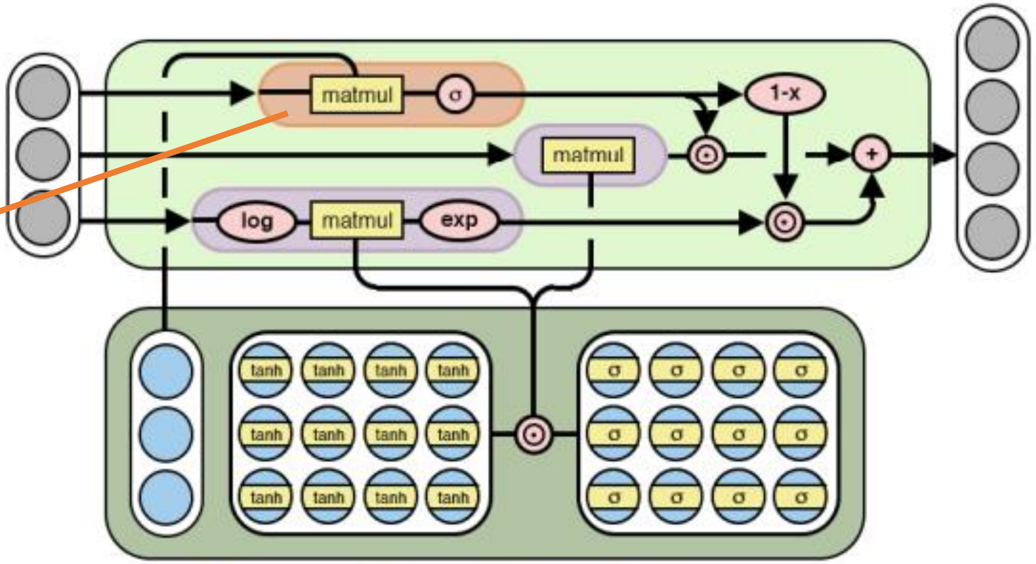
$$\mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

$$\texttt{NAC:} \quad \mathbf{a} = \mathbf{W}\mathbf{x}$$

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\boxed{\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})}$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

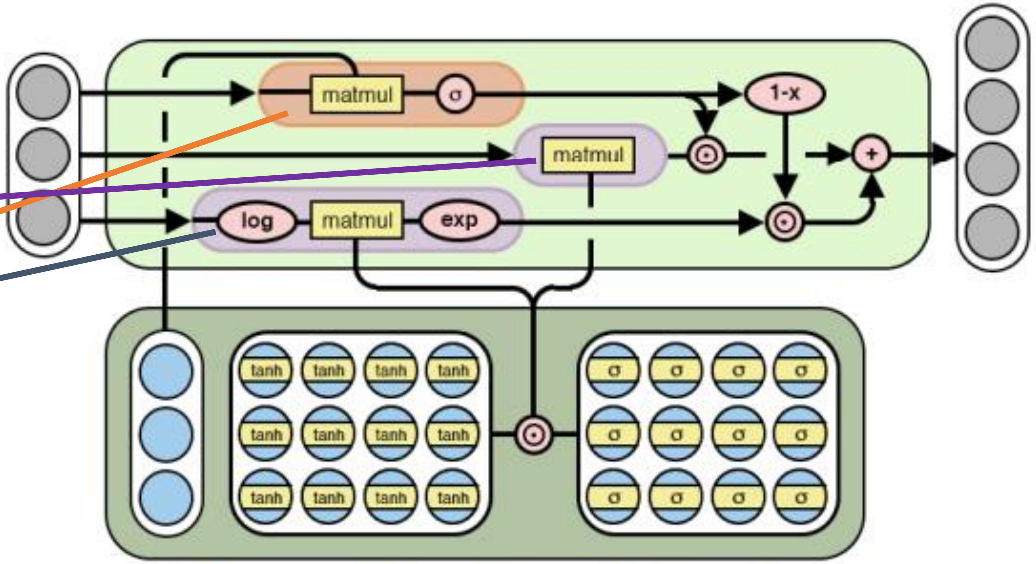
$$\mathbf{W} = \tanh(\hat{\mathbf{W}}) \odot \sigma(\hat{\mathbf{M}})$$

$$\texttt{NAC:} \quad \mathbf{a} = \mathbf{W}\mathbf{x}$$

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\mathbf{g} = \sigma(\mathbf{G}\mathbf{x})$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

To predict addition…

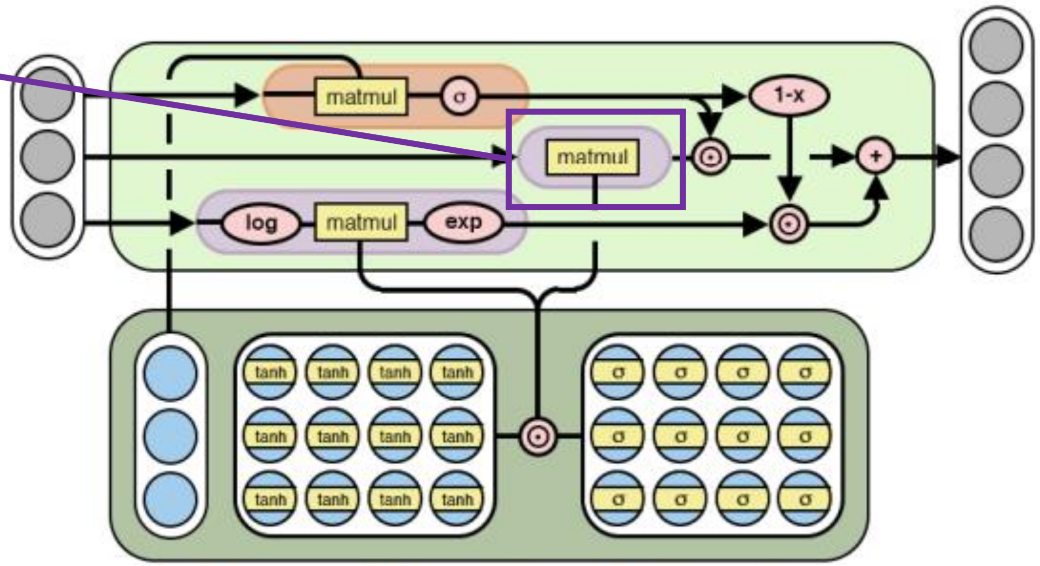
Ex) 3+5 = ?

x : [3 5]
W : [[1]
[1]] ← $\tanh(\hat{W}) \odot \mathrm{sigmoid}(\hat{M})$

a : Wx = [3 + 5] = [8]

NAC: $\mathbf{a} = \mathbf{W}\mathbf{x}$

NALU: $\mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$



(b) Neural Arithmetic Logic Unit (NALU)

To predict subtraction…

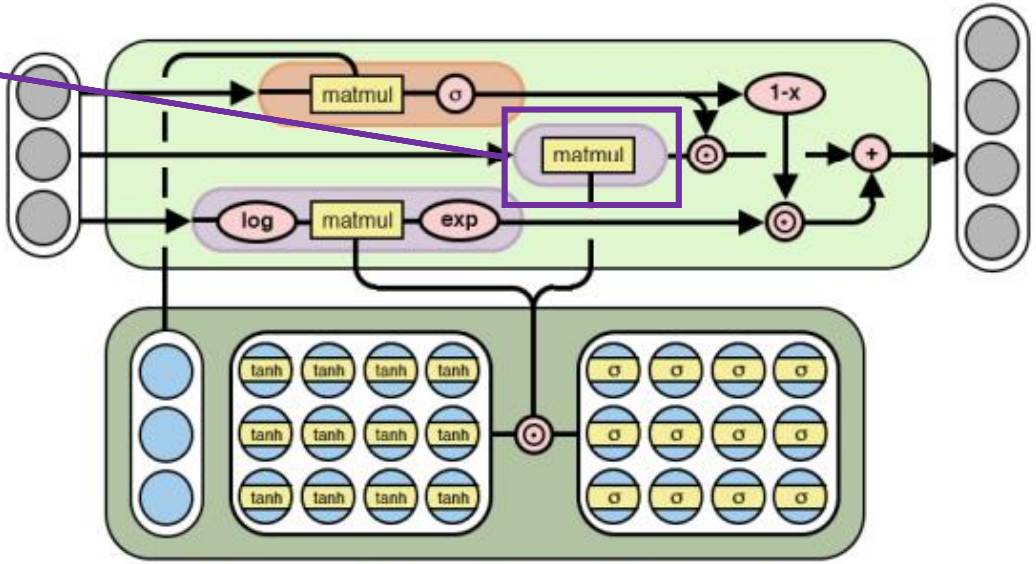
Ex) 3-5 = ?

x : [3 5]
W : [[1]
[-1]] ← tanh(Ŵ) ⊙ sigmoid(M̂)

a : Wx = [3 - 5] = [-2]

NAC: $\mathbf{a} = \mathbf{W}\mathbf{x}$

NALU: $\mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$



(b) Neural Arithmetic Logic Unit (NALU)

To predict multiplication…

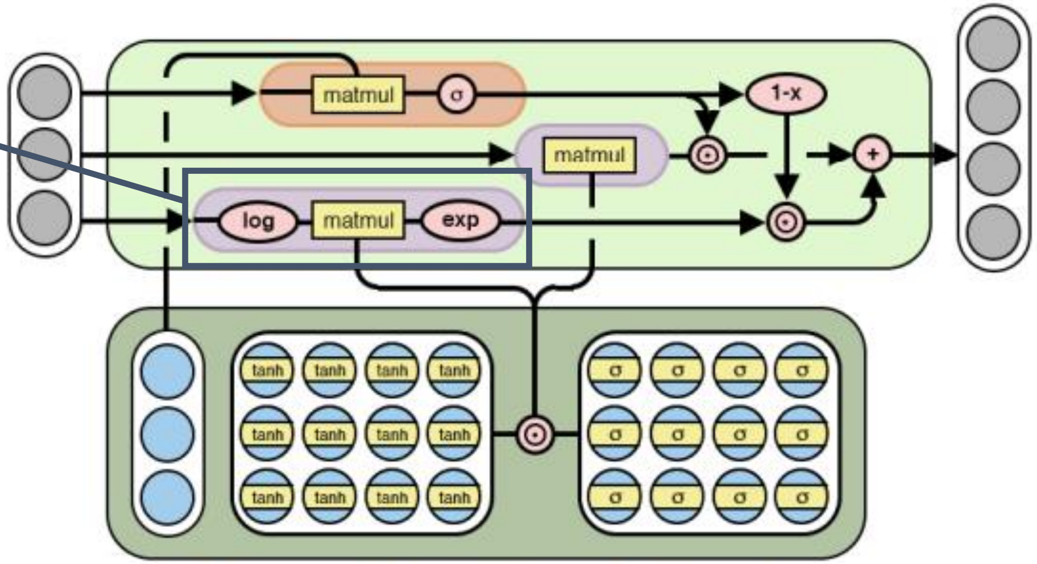
Ex) 3*5 = ?

x : [3 5]
W : [[1]
 [1]] ← tanh($\hat{W}$) ⊙ sigmoid($\hat{M}$)

W(log(|x|)) : [1*log|3| + 1*log|5|] = [log15]
exp(W(log(|x|)) : [exp(log15)] = [15]

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

To predict division…

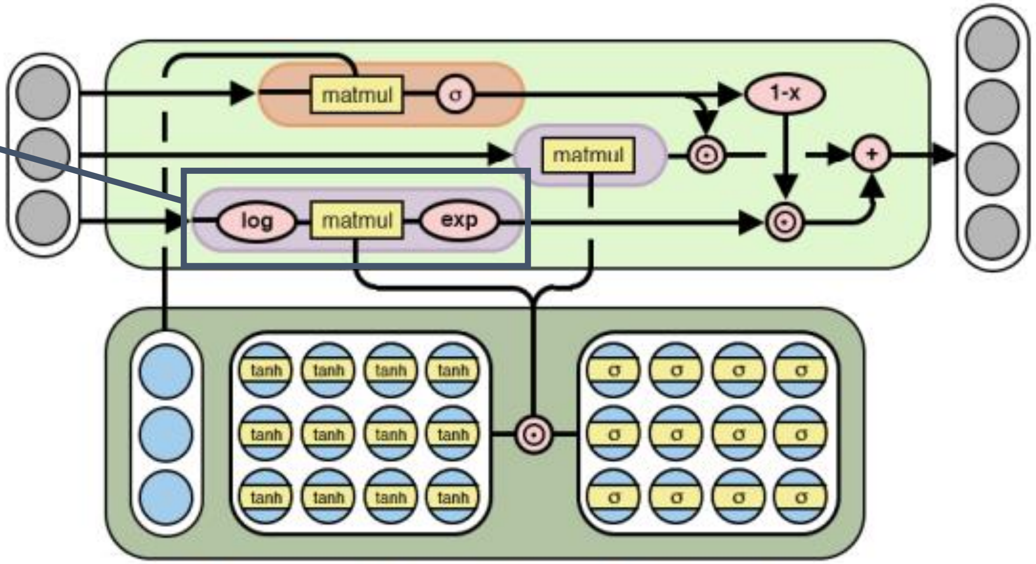
Ex) 3/5 = ?

x : [3 5]
W : [[1]
[-1]] ← tanh(Ŵ) ⊙ sigmoid(M̂)

W(log(|x|)) : [1*log|3| - 1*log|5|] = [log3/5]
exp(W(log(|x|)) : [exp(log3/5)] = [3/5]

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

To predict power function…

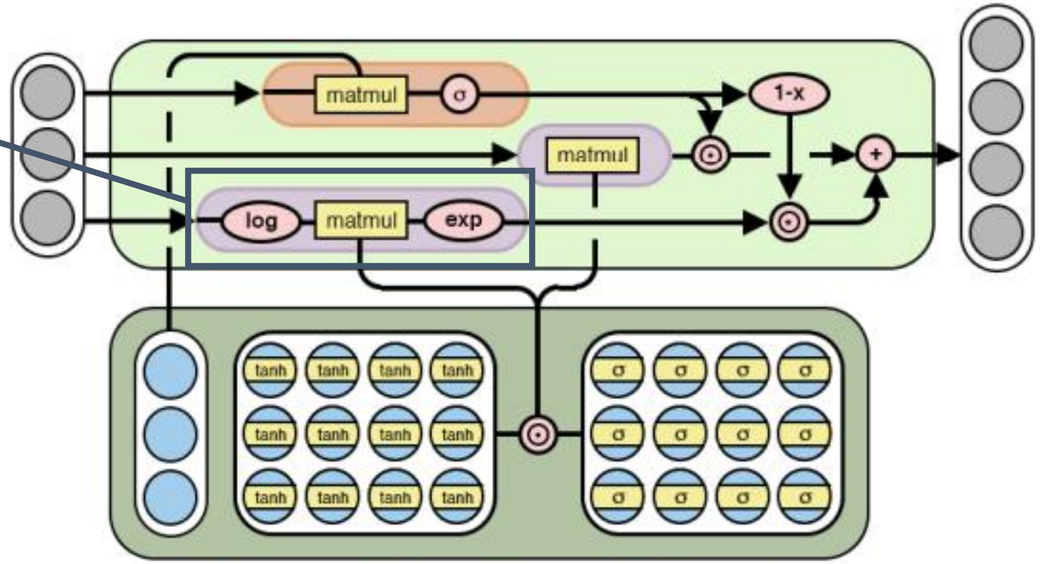
Ex) 3^1/2 = ?

x : [3]
W : [-0.5] ← tanh(Ŵ) ⊙ sigmoid(M̂)

W(log(|x|)) : [-0.5*log3] = [log(3^1/2)]
exp(W(log(|x|)) :[3^1/2]

$$\mathbf{m} = \exp \mathbf{W}(\log(|\mathbf{x}| + \epsilon))$$

$$\texttt{NALU:} \quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$

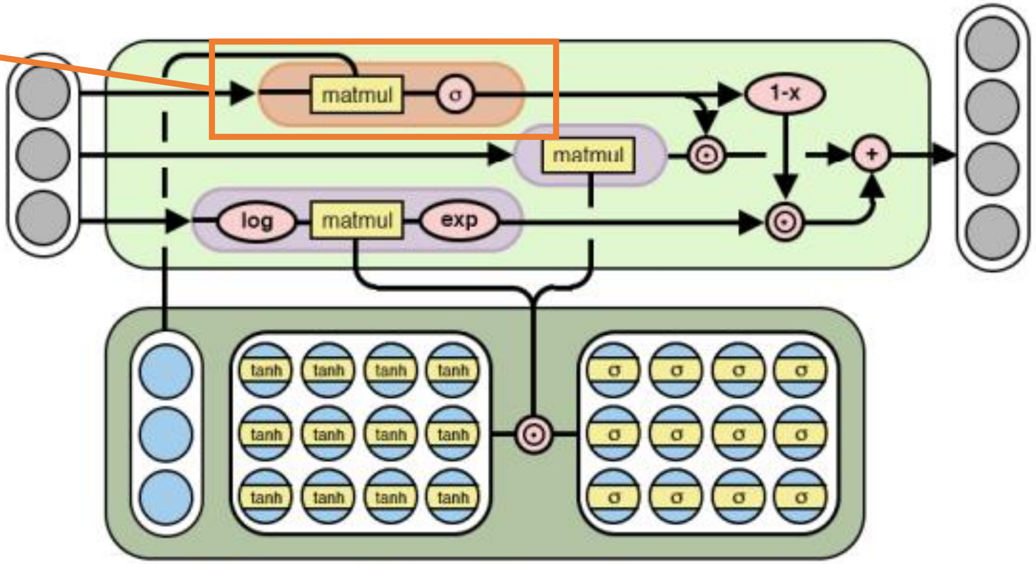


(b) Neural Arithmetic Logic Unit (NALU)

To select what to do…

Learned gate : select add/sub or mul/div

$$\texttt{NALU:}\quad \mathbf{y} = \mathbf{g} \odot \mathbf{a} + (1 - \mathbf{g}) \odot \mathbf{m}$$



(b) Neural Arithmetic Logic Unit (NALU)

Limitations of a single cell NALU

덧셈, 뺄셈 / 곱하기, 나누기 중 하나의 연산만 사용할 수 있다.

음수 관련 문제

Power operation의 범위는 지수가 [0, 1] 이내여야 한다.

Advantages of using NALU

수학적으로 정교하게 설계하여 interpolation & extrapolation 이 가능하다.

Static : input 전체가 한번에 들어옴 Recurrent : input이 시간에 걸쳐 순차적으로 들어옴

To select what to do…

		Static Task (test)				Recurrent Task (test)			
		Relu6	None	NAC	NALU	LSTM	ReLU	NAC	NALU
Interpolation	$a+b$	0.2	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	$a-b$	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	$a \times b$	3.2	20.9	21.4	**0.0**	**0.0**	**0.0**	1.5	**0.0**
	a/b	**4.2**	35.0	37.1	5.3	**0.0**	**0.0**	1.2	**0.0**
	a^2	0.7	4.3	22.4	**0.0**	**0.0**	**0.0**	2.3	**0.0**
	$\sqrt{a}$	0.5	2.2	3.6	**0.0**	**0.0**	**0.0**	2.1	**0.0**
Extrapolation	$a+b$	42.6	**0.0**	**0.0**	**0.0**	96.1	85.5	**0.0**	**0.0**
	$a-b$	29.0	**0.0**	**0.0**	**0.0**	97.0	70.9	**0.0**	**0.0**
	$a \times b$	10.1	29.5	33.3	**0.0**	98.2	97.9	88.4	**0.0**
	a/b	37.2	52.3	61.3	**0.7**	**95.6**	863.5	>999	>999
	a^2	47.0	25.1	53.3	**0.0**	98.0	98.0	123.7	**0.0**
	$\sqrt{a}$	10.3	20.0	16.4	**0.0**	95.8	34.1	>999	**0.0**

Table 1: Interpolation and extrapolation error rates for static and recurrent tasks. Scores are scaled relative to a randomly initialized model for each task such that 100.0 is equivalent to random, 0.0 is perfect accuracy, and >100 is worse than a randomly initialized model. Raw scores in Appendix B.

Table 1 summarizes results and shows that while several standard architectures succeed at these tasks in the interpolation case, none of them succeed at extrapolation. However, in both interpolation and extrapolation, the NAC succeeds at modeling addition and subtraction, whereas the more flexible NALU succeeds at multiplicative operations as well (except for division in the recurrent task[2]).

Model	Train MAE	Validation MAE	Test MAE
LSTM	0.003	29.9	29.4
LSTM + NAC	80.0	114.1	114.3
LSTM + NALU	0.12	**0.39**	**0.41**

Table 3: Mean absolute error (MAE) comparison on translating number strings to scalars. LSTM + NAC/NALU means a single LSTM layer followed by NAC or NALU, respectively.

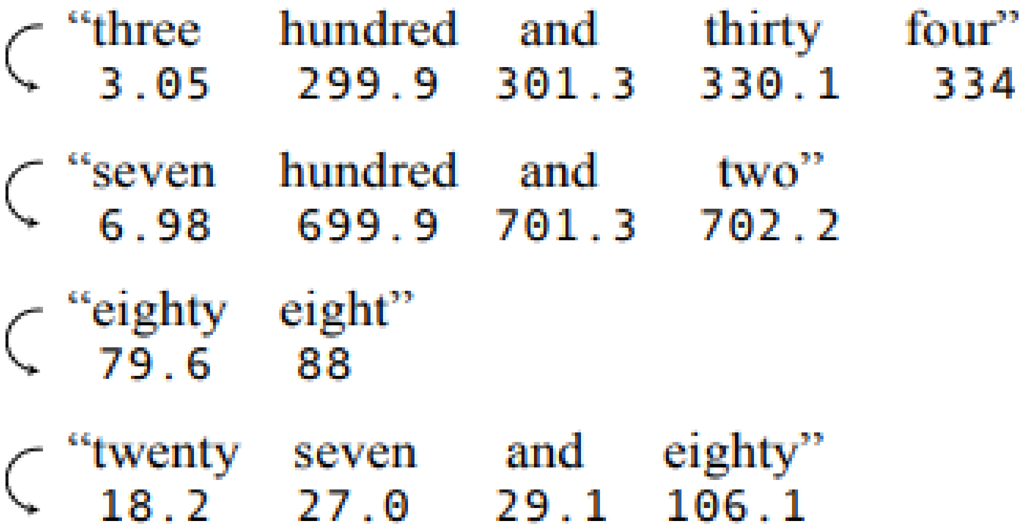
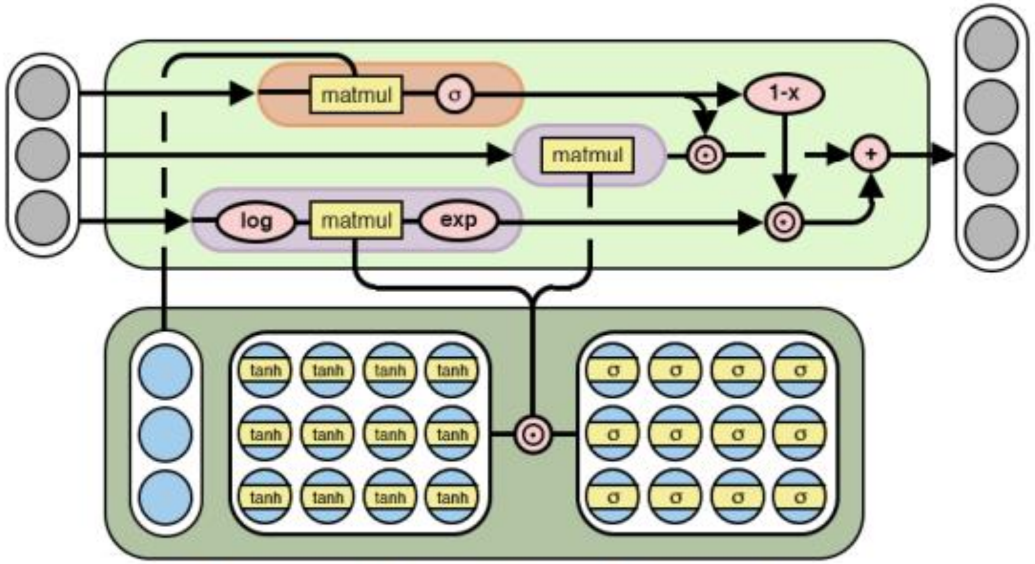


Figure 3: Intermediate NALU predictions on previously unseen queries.

DOOSAN dataset

I/O	Feature 수	Data 수
Ingot 중량 단조중량 최저진공도 작업시간 ⋮ SN H N O	Input Feature : 28개	113k개
인장강도 항복강도 상온충격	Output Feature : 3개	

Input(28)



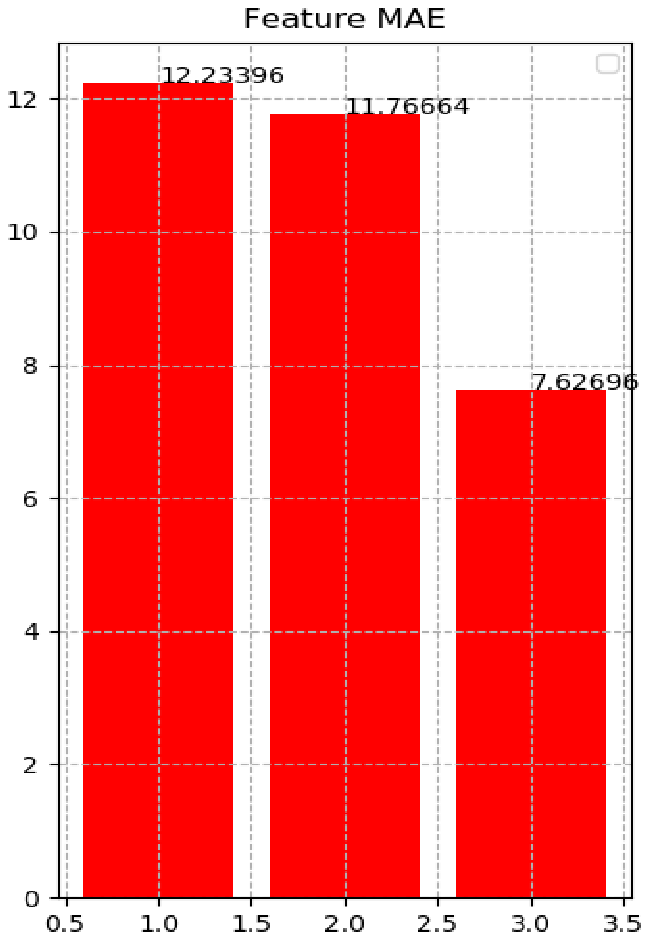
(b) Neural Arithmetic Logic Unit (NALU)

Output(3)

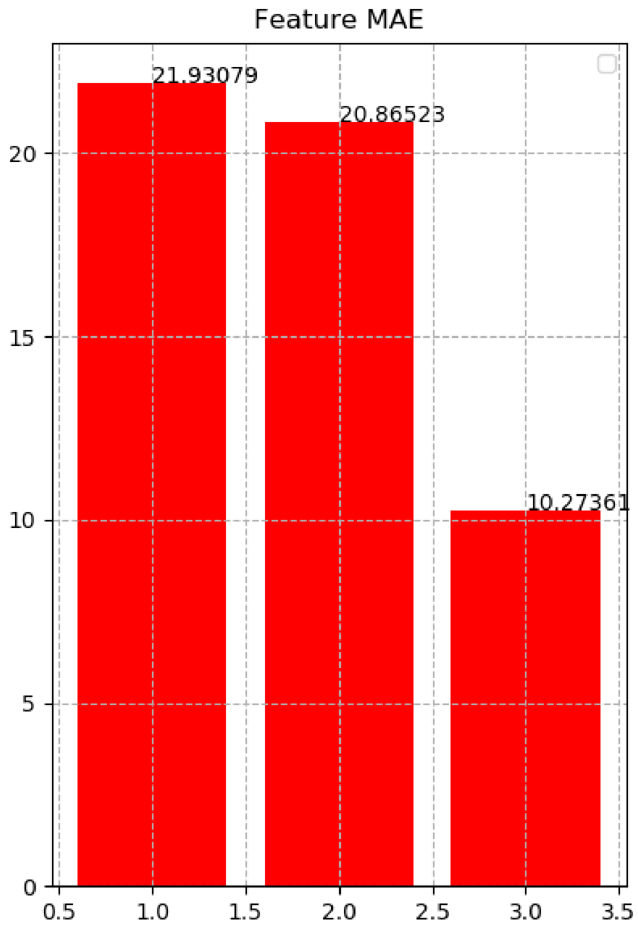
Baseline MAE error (MLP)

Feature 1: 16.467
Feature 2: 15.583
Feature 3: 8.900

RBF network



1 NALU cell
28 in – 3 out



2 NALU cells
28 in – 64 - 3 out