

CALU

: **C**omplex-Valued Neural **A**rithmetic **L**ogic **U**nit
with Neural Quantizer
for the Abstraction of Physical Symmetries in the
Nature

Wonsang Cho
(Seoul National University)

in collaboration with
Kayoung Ban, Dongsub Lee, Sungyeop Lee, Chanju Park

Summer Institute in Gangneung
2019 Aug 20

Supplimentary slides to K. Ban's talk today (Tue)

What and why do we do?

Methods of Mathematical Modeling

Models	Statistical/Machine Learning	Physics
Object Function	$E(f_k(f_j(..f_0(x w_0).. w_j) w_k))$	$\mathcal{L}(\phi(x), \dot{\phi} \alpha)$
Dynamical E.O.M	$\min_w E$	$\delta\mathcal{L} = 0$
Dynamical Index	data feeding sequence	time
Rep. (dynamical d.o.f)	neurons in w space, $f_i(x w)$	fields, $\phi_i(x)$ with spin
Rep. (master obj. function)	connections of neurons	operators under symmetries
Interactions	by regularizations	with coupled mediators
Model Parameters	Many, W(connec. weights)	Few, α (coupl.),...
Model Capacity	Flexible (for any system)	Limited (for effective system)
Interpretability (1 dyn. obj)	not interpretable	physical/meaningful
Interpretability (1 par.)	not interpretable	physical/meaningful
Model Search	data-driven	principle/symmetry-driven
Role of Constraints/Sym./Reg.	secondary	primary
Validity of Model	in x_{training}	in all x of eff. system (physics law)

← description of Machine Learnings as deep learnings

$$\begin{aligned}
\mathcal{L}_{SM} = & -\frac{1}{2}\partial_\nu g_\mu^a \partial_\nu g_\mu^a - g_s f^{abc} \partial_\mu g_\nu^a g_\mu^b g_\nu^c - \frac{1}{4}g_s^2 f^{abc} f^{ade} g_\mu^b g_\nu^c g_\mu^d g_\nu^e - \partial_\nu W_\mu^+ \partial_\nu W_\mu^- - \\
& M^2 W_\mu^+ W_\mu^- - \frac{1}{2}\partial_\nu Z_\mu^0 \partial_\nu Z_\mu^0 - \frac{1}{2c_w^2} M^2 Z_\mu^0 Z_\mu^0 - \frac{1}{2}\partial_\mu A_\nu \partial_\mu A_\nu - igc_w (\partial_\nu Z_\mu^0 (W_\mu^+ W_\nu^- - \\
& W_\nu^+ W_\mu^-) - Z_\nu^0 (W_\mu^+ \partial_\nu W_\mu^- - W_\mu^- \partial_\nu W_\mu^+) + Z_\mu^0 (W_\nu^+ \partial_\nu W_\mu^- - W_\nu^- \partial_\nu W_\mu^+)) - \\
& igs_w (\partial_\nu A_\mu (W_\mu^+ W_\nu^- - W_\nu^+ W_\mu^-) - A_\nu (W_\mu^+ \partial_\nu W_\mu^- - W_\mu^- \partial_\nu W_\mu^+) + A_\mu (W_\nu^+ \partial_\nu W_\mu^- - \\
& W_\nu^- \partial_\nu W_\mu^+)) - \frac{1}{2}g^2 W_\mu^+ W_\mu^- W_\nu^+ W_\nu^- + \frac{1}{2}g^2 W_\mu^+ W_\nu^- W_\mu^+ W_\nu^- + g^2 c_w^2 (Z_\mu^0 W_\mu^+ Z_\nu^0 W_\nu^- - \\
& Z_\mu^0 Z_\nu^0 W_\mu^+ W_\nu^-) + g^2 s_w^2 (A_\mu W_\mu^+ A_\nu W_\nu^- - A_\mu A_\nu W_\mu^+ W_\nu^-) + g^2 s_w c_w (A_\mu Z_\nu^0 (W_\mu^+ W_\nu^- - \\
& W_\nu^+ W_\mu^-) - 2A_\mu Z_\mu^0 W_\nu^+ W_\nu^-) - \frac{1}{2}\partial_\mu H \partial_\mu H - 2M^2 \alpha_h H^2 - \partial_\mu \phi^+ \partial_\mu \phi^- - \frac{1}{2}\partial_\mu \phi^0 \partial_\mu \phi^0 - \\
& \beta_h \left(\frac{2M^2}{g^2} + \frac{2M}{g} H + \frac{1}{2}(H^2 + \phi^0 \phi^0 + 2\phi^+ \phi^-) \right) + \frac{2M^4}{g^2} \alpha_h - \\
& g\alpha_h M (H^3 + H\phi^0 \phi^0 + 2H\phi^+ \phi^-) - \\
& \frac{1}{8}g^2 \alpha_h (H^4 + (\phi^0)^4 - 4(\phi^+ \phi^-)^2 + 4(\phi^0)^2 \phi^+ \phi^- + 4H^2 \phi^+ \phi^- + 2(\phi^0)^2 H^2) -
\end{aligned}$$

Great Success of Physics ...

This 1 page SM can predict a huge number of phenomena in weak

scale !

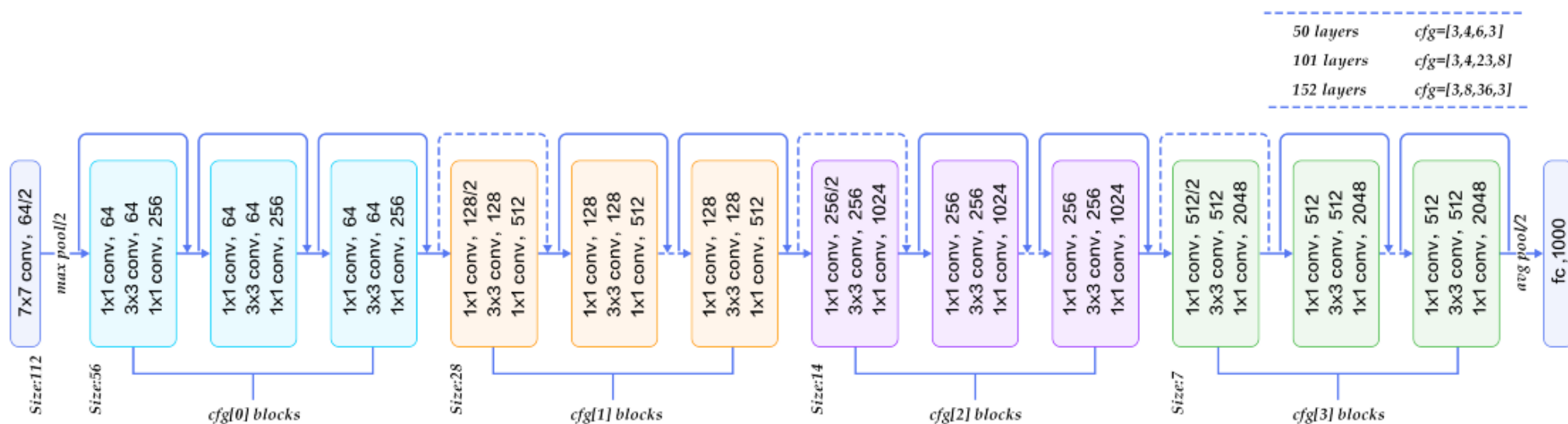
[in PDG book, see H. Murayama's talk (Sun)]

$$\begin{aligned}
& \frac{1}{2}g (W_\mu^+ (H\partial_\mu \phi^- - \phi^- \partial_\mu H) + W_\mu^- (H\partial_\mu \phi^+ - \phi^+ \partial_\mu H)) + \frac{1}{2}g \frac{1}{c_w} (Z_\mu^0 (H\partial_\mu \phi^0 - \phi^0 \partial_\mu H) + \\
& M (\frac{1}{2}Z_\mu^0 \partial_\mu \phi^0 + W_\mu^+ \partial_\mu \phi^- + W_\mu^- \partial_\mu \phi^+)) - ig \frac{s_w^2}{2c_w} M Z_\mu^0 (W_\mu^+ \phi^- - W_\mu^- \phi^+) + igs_w M A_\mu (W_\mu^+ \phi^- - \\
& W_\mu^- \phi^+) - \frac{1}{4}g^2 W_\mu^+ W_\mu^- (H^2 + (\phi^0)^2 + 2\phi^+ \phi^-) - \frac{1}{2}g^2 \frac{1}{c_w} Z_\mu^0 Z_\nu^0 (H^2 + (\phi^0)^2 + 2(2s_w^2 - 1)^2 \phi^+ \phi^-) - \\
& \frac{1}{2}g^2 \frac{s_w^2}{c_w} Z_\mu^0 \phi^0 (W_\mu^+ \phi^- + W_\mu^- \phi^+) - \frac{1}{2}g^2 \frac{s_w^2}{c_w} (W_\mu^+ \phi^- - W_\mu^- \phi^+) + \frac{1}{2}g^2 s_w A_\mu \phi^0 (W_\mu^+ \phi^- + \\
& W_\mu^- \phi^+) + \frac{1}{2}g^2 s_w A_\mu H (W_\mu^+ \phi^- - W_\mu^- \phi^+) - g^2 s_w (2c_w^2 - 1) Z_\mu^0 \phi^0 (\partial_\mu \phi^+ - \partial_\mu \phi^-) + \\
& m_u^\lambda u_j^\lambda - \bar{d}_j^\lambda (\gamma \partial + m_d^\lambda) d_j^\lambda + igs_w A_\mu (-\bar{e}^\lambda \gamma^\mu e^\lambda) + \frac{2}{3}(\bar{u}_j^\lambda \gamma^\mu u_j^\lambda) - \frac{1}{3}(\bar{d}_j^\lambda \gamma^\mu d_j^\lambda) + \\
& \frac{ig}{4c_w} Z_\mu^0 \{ (\bar{\nu}^\lambda \gamma^\mu (1 + \gamma^5) \nu^\lambda) + (\bar{e}^\lambda \gamma^\mu (4s_w^2 - 1 - \gamma^5) e^\lambda) + (\bar{d}_j^\lambda \gamma^\mu (\frac{4}{3}s_w^2 - 1 - \gamma^5) d_j^\lambda) + \\
& (\bar{u}_j^\lambda \gamma^\mu (1 - \frac{8}{3}s_w^2 + \gamma^5) u_j^\lambda) \} + \frac{ig}{2\sqrt{2}} W_\mu^+ ((\bar{\nu}^\lambda \gamma^\mu (1 + \gamma^5) U^{lep}{}_{\lambda\kappa} e^\kappa) + (\bar{u}_j^\lambda \gamma^\mu (1 + \gamma^5) C_{\lambda\kappa} d_j^\kappa)) + \\
& \frac{ig}{2\sqrt{2}} W_\mu^- ((\bar{e}^\kappa U^{lep}{}_{\kappa\lambda} \gamma^\mu (1 + \gamma^5) \nu^\lambda) + (\bar{d}_j^\kappa C_{\kappa\lambda}^\dagger \gamma^\mu (1 + \gamma^5) u_j^\lambda)) + \\
& \frac{ig}{2M\sqrt{2}} \phi^+ (-m_e^\kappa (\bar{\nu}^\lambda U^{lep}{}_{\lambda\kappa} (1 - \gamma^5) e^\kappa) + m_\nu^\lambda (\bar{\nu}^\lambda U^{lep}{}_{\lambda\kappa} (1 + \gamma^5) e^\kappa) + \\
& \frac{ig}{2M\sqrt{2}} \phi^- (m_e^\lambda (\bar{e}^\lambda U^{lep}{}_{\lambda\kappa}^\dagger (1 + \gamma^5) \nu^\kappa) - m_\nu^\nu (\bar{e}^\lambda U^{lep}{}_{\lambda\kappa}^\dagger (1 - \gamma^5) \nu^\kappa) - \frac{g}{2} \frac{m_\nu^\lambda}{M} H (\bar{\nu}^\lambda \nu^\lambda) - \\
& \frac{g}{2} \frac{m_\lambda^\lambda}{M} H (\bar{e}^\lambda e^\lambda) + \frac{ig}{2} \frac{m_\lambda^\lambda}{M} \phi^0 (\bar{\nu}^\lambda \gamma^5 \nu^\lambda) - \frac{ig}{2} \frac{m_\lambda^\lambda}{M} \phi^0 (\bar{e}^\lambda \gamma^5 e^\lambda) - \frac{1}{4} \bar{\nu}_\lambda M_{\lambda\kappa}^R (1 - \gamma_5) \hat{\nu}_\kappa - \\
& \frac{1}{4} \bar{\nu}_\lambda M_{\lambda\kappa}^R (1 - \gamma_5) \hat{\nu}_\kappa + \frac{ig}{2M\sqrt{2}} \phi^+ (-m_d^\kappa (\bar{u}_j^\lambda C_{\lambda\kappa} (1 - \gamma^5) d_j^\kappa) + m_u^\lambda (\bar{u}_j^\lambda C_{\lambda\kappa} (1 + \gamma^5) d_j^\kappa) + \\
& \frac{ig}{2M\sqrt{2}} \phi^- (m_d^\lambda (\bar{d}_j^\lambda C_{\lambda\kappa}^\dagger (1 + \gamma^5) u_j^\kappa) - m_u^\kappa (\bar{d}_j^\lambda C_{\lambda\kappa}^\dagger (1 - \gamma^5) u_j^\kappa) - \frac{g}{2} \frac{m_\lambda^\lambda}{M} H (\bar{u}_j^\lambda u_j^\lambda) - \\
& \frac{g}{2} \frac{m_\lambda^\lambda}{M} H (\bar{d}_j^\lambda d_j^\lambda) + \frac{ig}{2} \frac{m_\lambda^\lambda}{M} \phi^0 (\bar{u}_j^\lambda \gamma^5 u_j^\lambda) - \frac{ig}{2} \frac{m_\lambda^\lambda}{M} \phi^0 (\bar{d}_j^\lambda \gamma^5 d_j^\lambda) + \bar{G}^a \partial^2 G^a + g_s f^{abc} \partial_\mu \bar{G}^a G^b g_\mu^c + \\
& \bar{X}^+ (\partial^2 - M^2) X^+ + \bar{X}^- (\partial^2 - M^2) X^- + \bar{X}^0 (\partial^2 - \frac{M^2}{c_w^2}) X^0 + \bar{Y} \partial^2 Y + igc_w W_\mu^+ (\partial_\mu \bar{X}^0 X^- - \\
& \partial_\mu \bar{X}^+ X^0) + igs_w W_\mu^+ (\partial_\mu \bar{Y} X^- - \partial_\mu \bar{X}^+ Y) + igc_w W_\mu^- (\partial_\mu \bar{X}^- X^0 - \\
& \partial_\mu \bar{X}^0 X^+) + igs_w W_\mu^- (\partial_\mu \bar{X}^- Y - \partial_\mu \bar{Y} X^+) + igc_w Z_\mu^0 (\partial_\mu \bar{X}^+ X^+ - \\
& \partial_\mu \bar{X}^- X^-) + igs_w A_\mu (\partial_\mu \bar{X}^+ X^+ - \\
& \partial_\mu \bar{X}^- X^-) - \frac{1}{2}gM (\bar{X}^+ X^+ H + \bar{X}^- X^- H + \frac{1}{c_w^2} \bar{X}^0 X^0 H) + \frac{1-2c_w^2}{2c_w} igM (\bar{X}^+ X^0 \phi^+ - \bar{X}^- X^0 \phi^-) + \\
& \frac{1}{2c_w} igM (\bar{X}^0 X^- \phi^+ - \bar{X}^0 X^+ \phi^-) + igM s_w (\bar{X}^0 X^- \phi^+ - \bar{X}^0 X^+ \phi^-) + \\
& \frac{1}{2}igM (\bar{X}^+ X^+ \phi^0 - \bar{X}^- X^- \phi^0) .
\end{aligned}$$

Success of Machine Learning ...

Image Recognition (ImageNet)

: Deep Neural Nets in $\sim O(10\text{GBytes})$ can predict the correct class of ~ 1 million images in total 1000 classes, with top-5 error less than $\sim 3\%$



Can we combine the advantages of the two?

→ How to create a NN architecture
which can encode physics with
elementary mathematical objects ?

Methods of Mathematical Modeling

Models	Statistical/Machine Learning	Physics
Object Function	$E(f_k(f_j(..f_0(x w_0).. w_j) w_k))$	$\mathcal{L}(\phi(x), \dot{\phi} \alpha)$
Dynamical E.O.M	$\min_w E$	$\delta\mathcal{L} = 0$
Dynamical Index	data feeding sequence	time
Re	<p>→ We want to build an NN architecture which can be trained data-driven way using back-propagation, so every neuron object can have its own physical abstraction by elementary functions.</p>	
Model Parameters	Many , W (connec. weights)	Few , α (coupl.),...
Model Capacity	Flexible (for any system)	Limited (for effective system)
Interpretability (1 dyn. obj)	not interpretable	physical/meaningful
Interpretability (1 par.)	not interpretable	physical/meaningful
Model Search	data-driven	principle/symmetry-driven
Role of Constraints/Sym./Reg.	secondary	primary
Validity of Model	in x_{training}	in all x of eff. system (physics law)

Failures of Deep Learning ...

Actually, usual DNN requires lots of data for here and there, ... to learn and build a good model toward the model in ground-truth, **without bias / overfitting**.

Such a strong data dependency of deep learning, may be a severe defect, if toward an AI architecture especially using the **data of physical science which possesses more robust mathematical relations inside**, and we believe that good ML models for physical science should be trained valid everywhere without entire data coverage, like as we believe that the physics laws which is discovered and proven to be valid here, is then valid everywhere, for a given energy or multiplicity scale.

→ **‘Universality of Scientific ML models’**

One tiny failure ...

(may be BIG for the universality of DL models)

→ DNNs cannot even learn simple arithmetic operations in a domain-region free way.

Exp) Identity Op ($\mathbf{x} \rightarrow \mathbf{x}$)

	tanh	ReLU
Interp. Error	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$
Extrap. Error	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$

Exp) Arithmetic Ops

		tanh	ReLU
Interpolation E	$x_1 + x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$
	$x_1 - x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$
	$x_1 * x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$
	x_1/x_2	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$
Extrapolation E	$x_1 + x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$
	$x_1 - x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$
	$x_1 * x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+6})$
	x_1/x_2	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+7})$

The model's universality cannot be improved by
increasing model's capacity /
by training-procedure optimization.

*It is a matter of architecture itself, especially on
how to encode non-linearities.*

Our approach (to control or improve the model's universality) is to embed the structure of 'function approximation in power series' inside the DNN, in a trainable way using back-propagation, while borrowing main non-linearities from the power series, rather than from the activations.

We can handle the errors in exterior region by increasing layer's power, and also by disconnecting unnecessary connections by exactly zero-weights quantized.

Actually, power series expansion is a conventional way for many modern computers to approximate the elementary functions – which are the basic building blocks for representing the physics law.

→ Why not in DNN for scientific data?

Architecture of CALU

(Complex-valued Neural **A**rithmetic **L**ogic **U**nit)
with Neural Quantizer

Exp1) learning basic arithmetic operations
(+, -, ×, ÷)

Exp2) learning polynomial functions

Related Works

1) Neural Arithmetic Logic Unit (NALU)

A. Trask, F. Hill, S. Reed, J. Rae, C. Dyer, P. Blunsom [arXiv:1808.00508]

2) Binary/Ternary connected networks

M. Courbariaux, Y. Bengio, Jean-Pierre David [arXiv:1511.00363]

Zhouhan Lin, M. Courbariaux, R. Memisevie, Y. Bengio [arXiv:1510.03009]

C. Zhu, S. Han, H. Mao, W. J. Dally [‘Trained Ternary Quantization’]

...

3) Complex-valued neural networks

T. Kim, T. Adah [‘Approximation by Fully Complex Multilayer Perceptrons’]

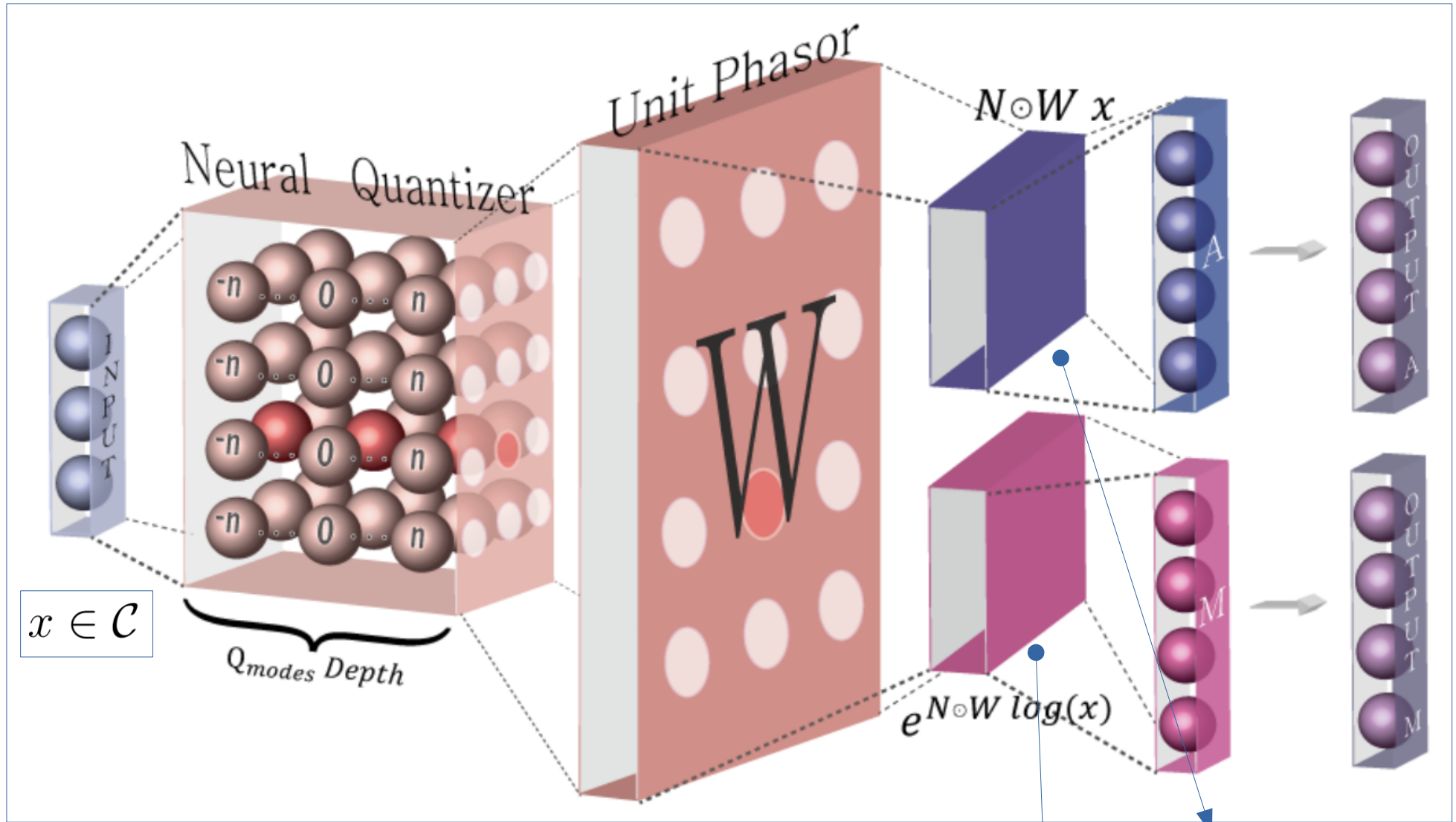
T. Nitta [‘An Extension of the BP algorithm to complex numbers’]

C. Trabelsi et. al. [‘Deep Complex Networks’]

N. Guberman [‘On Complex-Valued CNN’]

....

One CALU layer in a Complex-Valued Neural Network



$$N_{kj} = \sum_{n \in Q} \left[n \times \frac{\exp(\hat{N}_{kj,n})}{\sum_l \exp(\hat{N}_{kj,l})} \right]$$

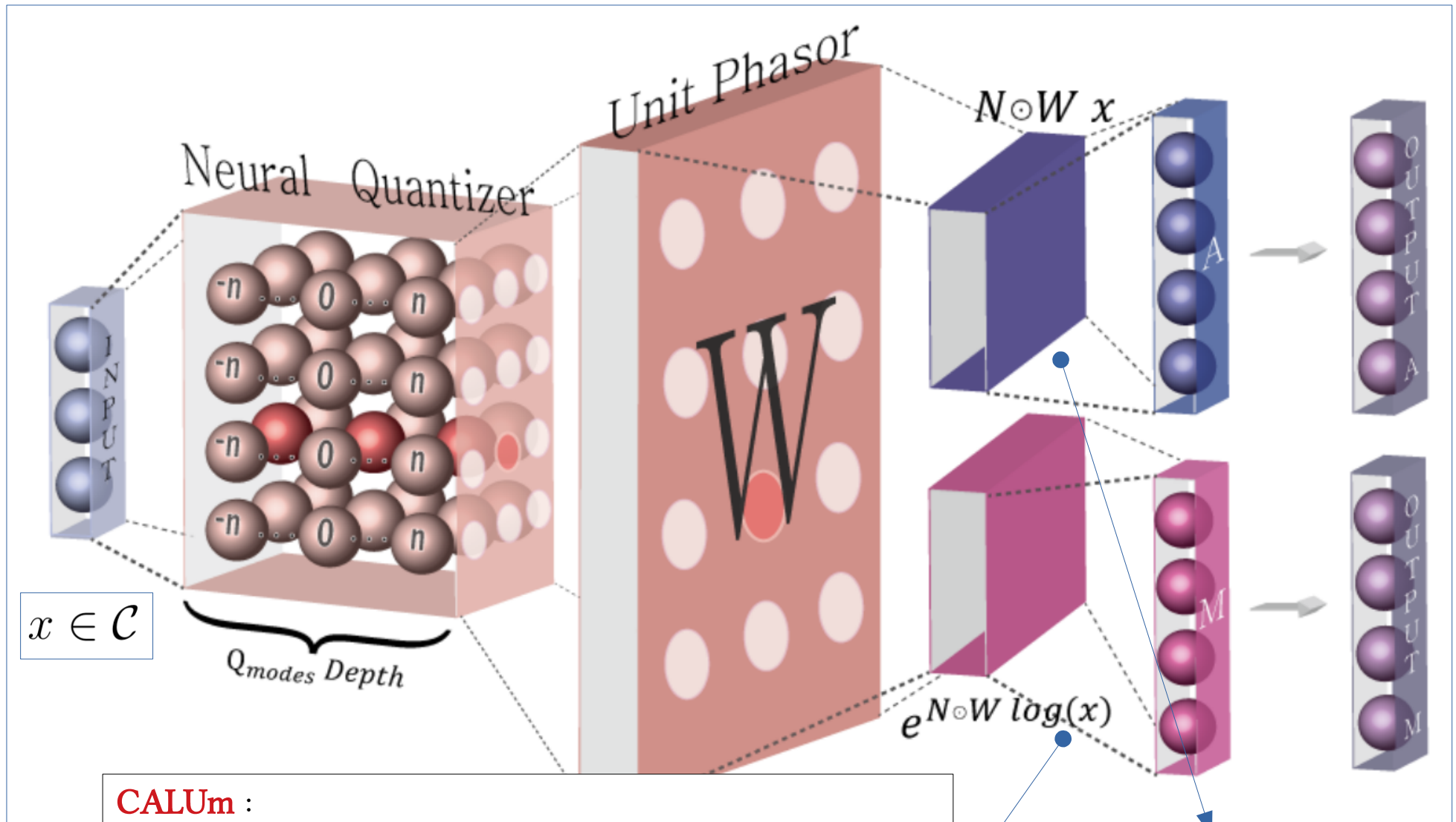
ex) $Q = \{-N_{max}, \dots, 0, \dots, N_{max}\}$

$$W = e^{+i\pi\sigma(\hat{\theta})}$$

CALUm :
for multiplication/div. with
(quantized) power op.

CALUa :
for addition/sub. with
(quantized) scaling

One CALU layer in a Complex-Valued Neural Network



CALUm :

'exp-add-log' structure with complex- $\log(x)$ in CVNN can enable us to do multiplication op. with the inputs of any phases, also with weights being consistently trainable using back-propagation of CVNN.

N_{kj}

CALUa :

for addition/sub. with (quantized) scaling

ex) $Q = \{-N_{max}, ..0, .., N_{max}\}$

Neural Quantizer, N

- Expectation value of picking a quantized value in a set Q , with each element n , weighted by the probability $P(n)$ modeled by softmax function with learnable parameters, N -hats for each n .
- Q is a set of quantized values (in general C), which can encode any values from dynamics.

ex) $Q = \{-2, -1, 0, 1, 2\}$ with the unit phasor $W = 1 \rightarrow$ R-valued total weights

ex) $Q = \{0, 1, 2, \dots, N_{\max}\}$ with $W = \exp [i \pi \sigma] \rightarrow$ C-valued total weights

- k, j : node indexes connecting layers
- N_{kj} is asymptotically stabilized and trained to be a Q_i , in back-propagation

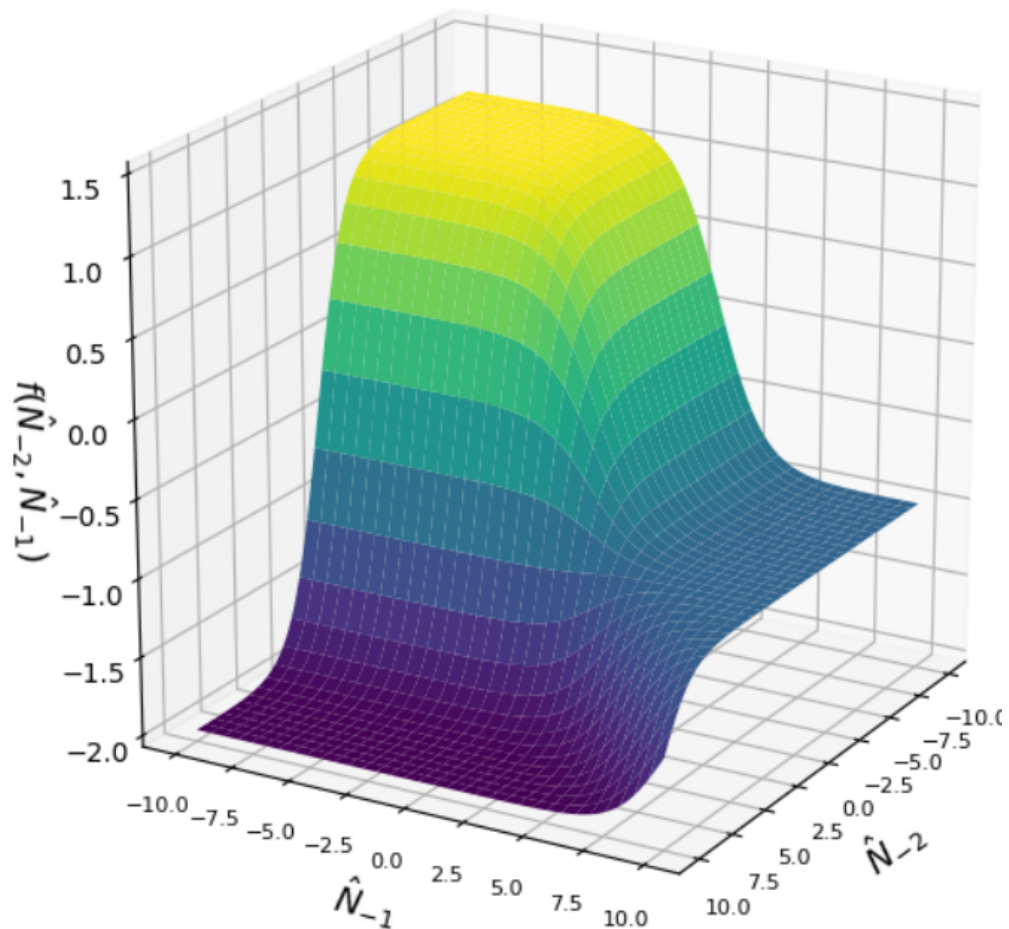
$$\frac{\partial N_{kj}}{\partial \hat{N}_{kj,n}} = -P_{kj,n}(N_{kj} - n), \quad P_{kj,n} = \frac{\exp(\hat{N}_{kj,n})}{\sum_l \exp(\hat{N}_{kj,l})}$$

\rightarrow coupled Boltzmann transport equation.

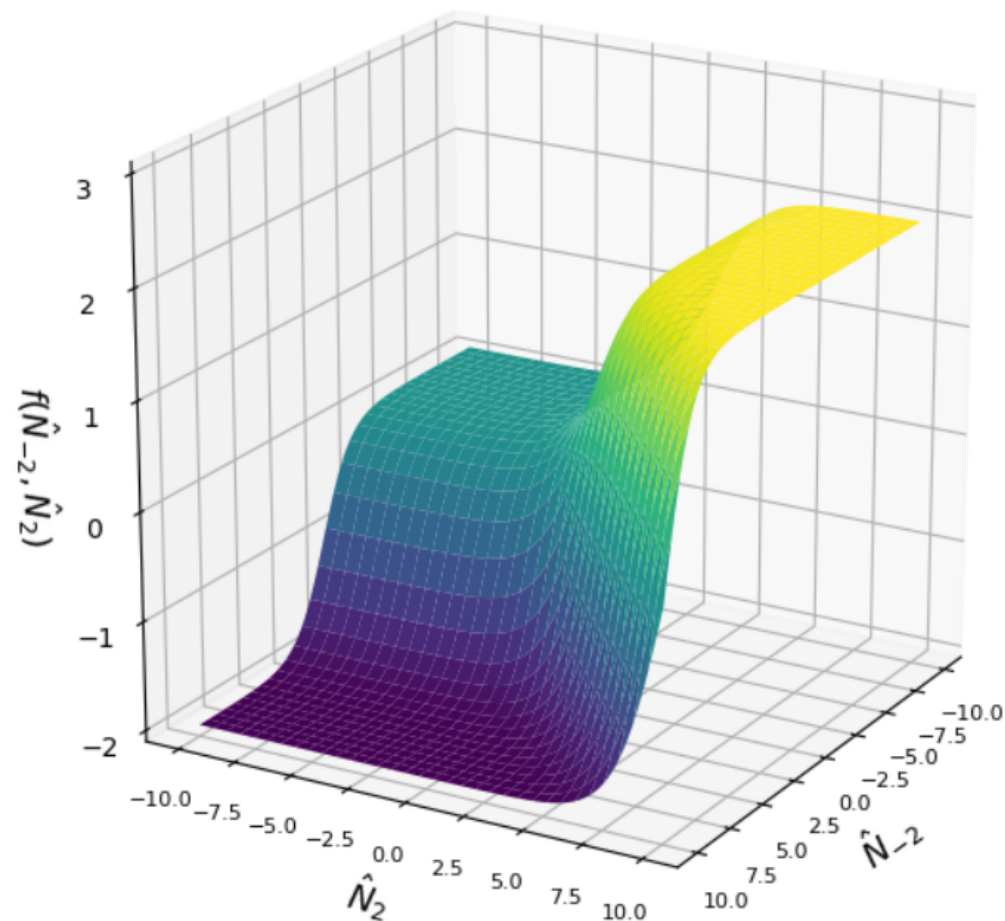
Behavior of Neural Quantizer, N

ex) $Q = \{-2, -1, 0, 1, 2\}$

-2 & -1 Quantizer



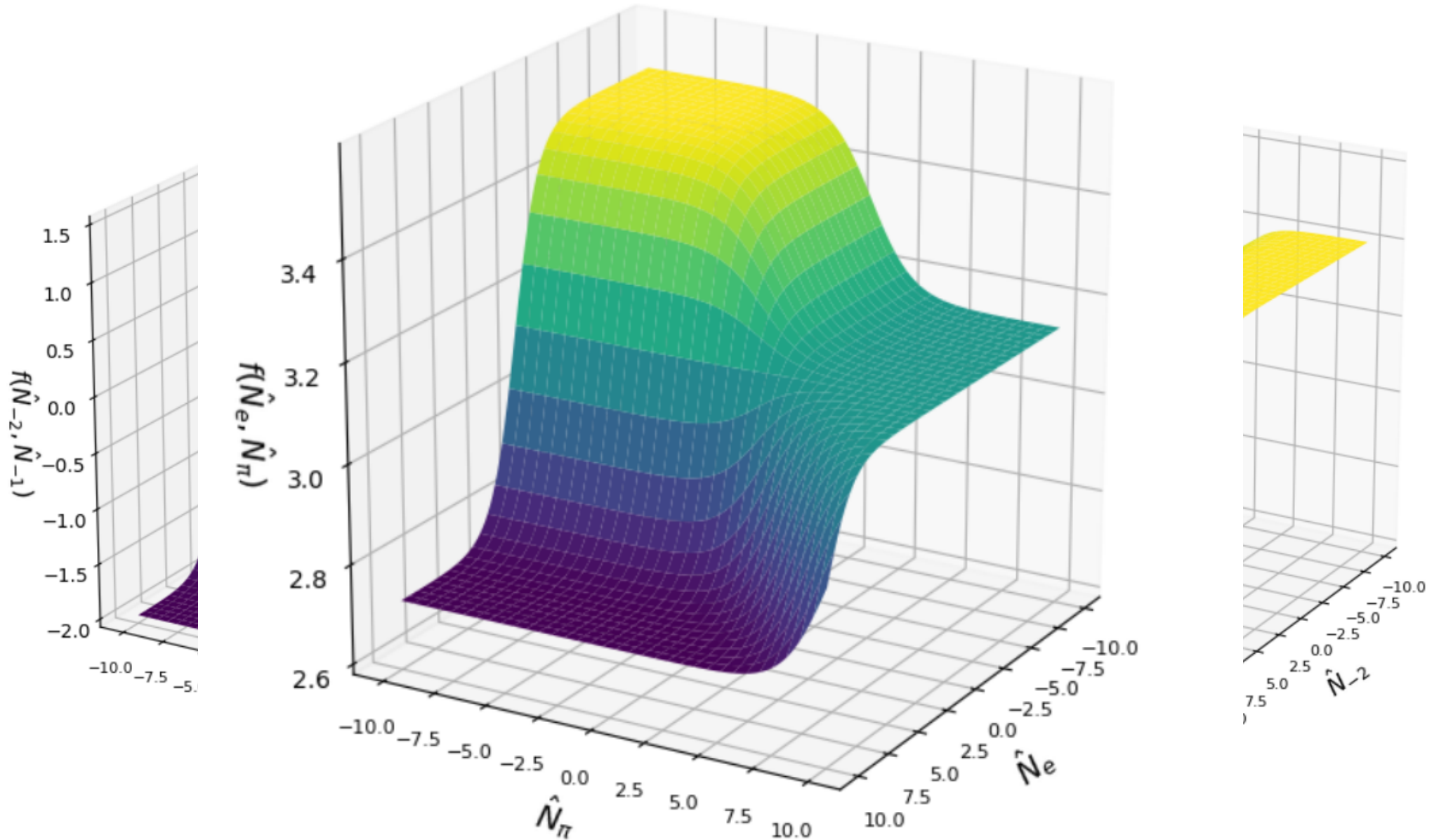
-2 & 2 Quantizer



Behavior of Neural Quantizer, N

ex)

e & π Quantizer



Unit Phasor

$$W_{kj} = e^{+i\pi\sigma(\hat{\theta}_{kj})}$$

- can be trained into the quantized phases ($\{0, \pi\}$)
- so the overall sign of weights $\{+1, -1\}$ can be trained in the complex-domain, with learnable parameters, theta-hats
- trainable, if Q consists of all positive numbers, otherwise just set to 1.

all in all ...

CALUa

- linear layer with scaling weights ($N \circ W$) quantized in Q for addition and subtraction.

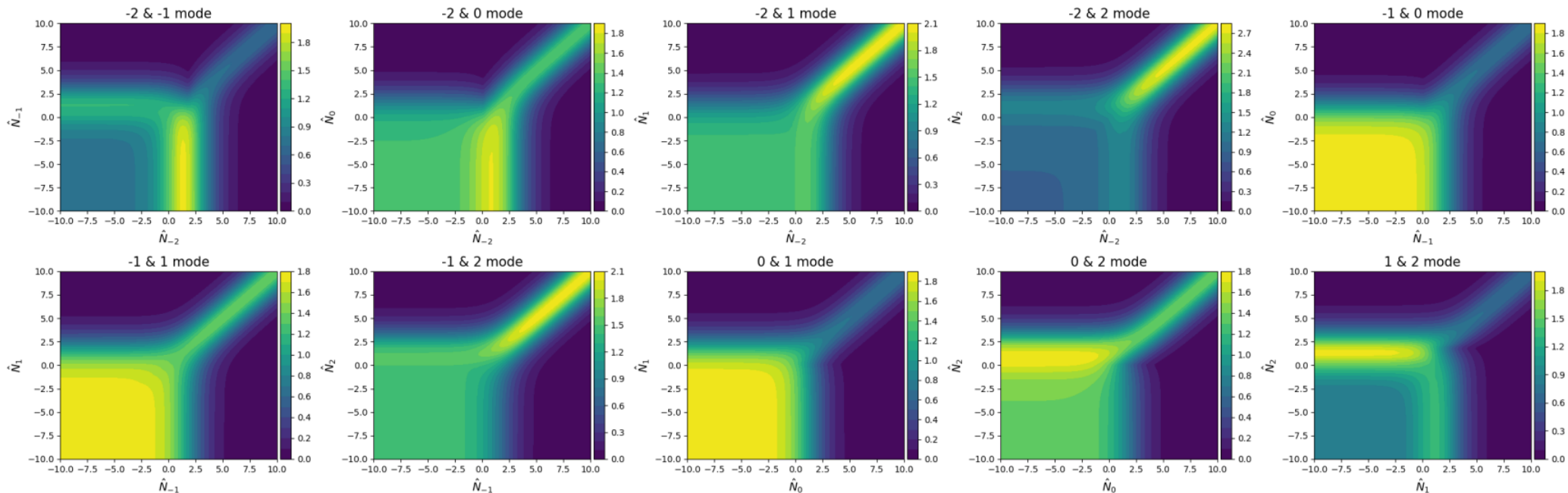
CALUm

- exp-add-log layer with powering weights ($N \circ W$) quantized in Q for mul. and div.
- **The first NN architecture** which can learn the multiplication and division op. of input variables **in general domain, trainable using BP, in precision with universality.**

Regularization Potential

$$E_{kj} = \sum_{n \in Q} |N_{kj} - n| \log [(1 - P_{kj,n})]$$

- for boosting the training of neural quantizer
- $E_{kj} \rightarrow 0$ for $N_{kj}=n$, also in preference of adjacent modes for jumping into.



Experiments

Experiment 1. Regression to a constant (1)

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

NN Model	Layer Structure
tanh	DIM(X) 100:TANH 100:TANH DIM(Y):IDENTITY
ReLU	DIM(X) 100:RELU 100:RELU DIM(Y):IDENTITY
NALU	DIM(X) 50:NALU _m DIM(Y):NALU _a
CALU	DIM(X) 50:CALU _m DIM(Y):CALU _a

Experiment 1. Regression to a constant (1)

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

	tanh	ReLU	NALU	CALU
Interp. Error	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathbf{\mathcal{O}(10^{-25})}$
Extrap. Error	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{-4})$	$< \mathbf{\mathcal{O}(10^{-21})}$

Experiment 2. Regression as identity

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix}$$

NN Model	Layer Structure
tanh	DIM(X) 100:TANH 100:TANH DIM(Y):IDENTITY
ReLU	DIM(X) 100:RELU 100:RELU DIM(Y):IDENTITY
NALU	DIM(X) 50:NALU _m DIM(Y):NALU _a
CALU	DIM(X) 50:CALU _m DIM(Y):CALU _a

Experiment 2. Regression as identity

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_8 \end{bmatrix}$$

	tanh	ReLU	NALU	CALU
Interp. Error	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathbf{\mathcal{O}(10^{-25})}$
Extrap. Error	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{-4})$	$< \mathbf{\mathcal{O}(10^{-21})}$

Experiment 3. Elementary Arithmetic Operations

$$\text{Addition} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1 + x_2 \end{bmatrix}$$

$$\text{Subtraction} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1 - x_2 \end{bmatrix}$$

$$\text{Multiplication} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1 * x_2 \end{bmatrix}$$

$$\text{Division} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} x_1/x_2 \end{bmatrix}$$

Experiment 3. Elementary Arithmetic Operations

NN Model	Layer Structure
tanh	DIM(X) 100:TANH 100:TANH DIM(Y):IDENTITY
ReLU	DIM(X) 100:RELU 100:RELU DIM(Y):IDENTITY
NALU	DIM(X) 5:NALU DIM(Y):NALU
CALU	DIM(X) 5:CALU DIM(Y):CALU

		tanh	ReLU	NALU	CALU
Interpolation E	$x_1 + x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-20})$
	$x_1 - x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-20})$
	$x_1 * x_2$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-2})$	$< \mathcal{O}(10^{-16})$
	x_1/x_2	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{+1})$	$< \mathcal{O}(10^{-16})$
Extrapolation E	$x_1 + x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{-2})$	$< \mathcal{O}(10^{-17})$
	$x_1 - x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{-2})$	$< \mathcal{O}(10^{-17})$
	$x_1 * x_2$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+6})$	$< \mathcal{O}(10^{+6})$	$< \mathcal{O}(10^{-14})$
	x_1/x_2	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+7})$	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{-15})$

Experiment 4. Multinomial Expansion

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} 1 \\ (x_1 + x_2 + x_3) \\ (x_1 + x_2 + x_3)^2 \\ (x_1 + x_2 + x_3)^3 \end{bmatrix}$$

NN Model	Layer Structure
tanh	DIM(X) 100:TANH 100:TANH DIM(Y):IDENTITY
ReLU	DIM(X) 100:RELU 100:RELU DIM(Y):IDENTITY
NALU	DIM(X) 100:NALU _m DIM(Y):NALU _a
CALU	DIM(X) 100:CALU _m DIM(Y):CALU _a

Experiment 4. Multinomial Expansion

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} 1 \\ (x_1 + x_2 + x_3) \\ (x_1 + x_2 + x_3)^2 \\ (x_1 + x_2 + x_3)^3 \end{bmatrix}$$

	tanh	ReLU	NALU	CALU
Interp. Error	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-2})$	$< \mathbf{\mathcal{O}(10^{-25})}$
Extrap. Error	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+6})$	$< \mathcal{O}(10^{+6})$	$< \mathbf{\mathcal{O}(10^{-20})}$

Experiment 5. Learning 3 Lorentz scalars in 2 four-vectors

$$\begin{bmatrix} p_0 \\ \vdots \\ p_3 \\ q_0 \\ \vdots \\ q_3 \end{bmatrix} \rightarrow \text{NN model} \rightarrow \begin{bmatrix} p \cdot p \\ p \cdot q \\ q \cdot q \end{bmatrix}$$

NN Model	Layer Structure
tanh	DIM(X) 100:TANH 100:TANH DIM(Y):IDENTITY
ReLU	DIM(X) 100:RELU 100:RELU DIM(Y):IDENTITY
NALU	DIM(X) 50:NALU _m DIM(Y):NALU _a
CALU	DIM(X) 50:CALU _m DIM(Y):CALU _a

Experiment 5. Learning 3 Lorentz scalars in 2 four-vectors

	tanh	ReLU	NALU	CALU
Interp. Error	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-6})$	$< \mathcal{O}(10^{-2})$	$< \mathbf{\mathcal{O}(10^{-20})}$
Extrap. Error	$< \mathcal{O}(10^{+3})$	$< \mathcal{O}(10^{+6})$	$< \mathcal{O}(10^{+6})$	$< \mathbf{\mathcal{O}(10^{-16})}$

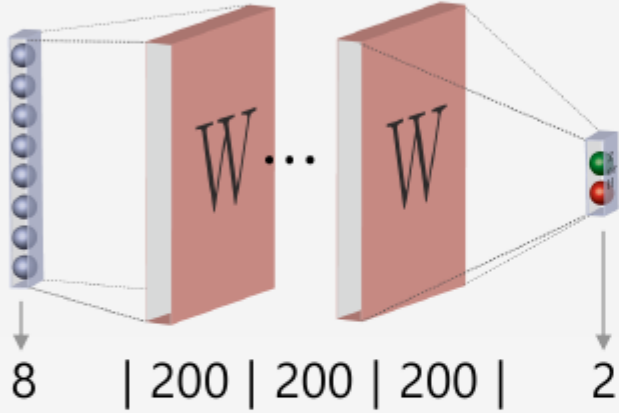
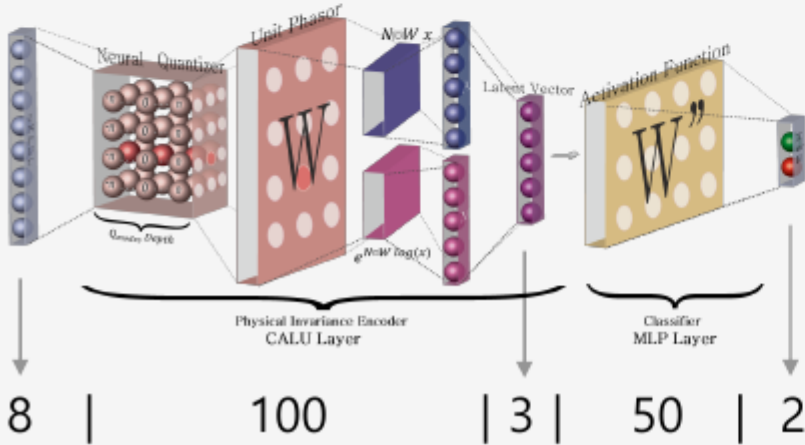
New classification model with new feature extraction by CALUs

with **Kayoung Ban (talk today)**, Sungyeop Lee, Chanju Park, and Seong Chan Park

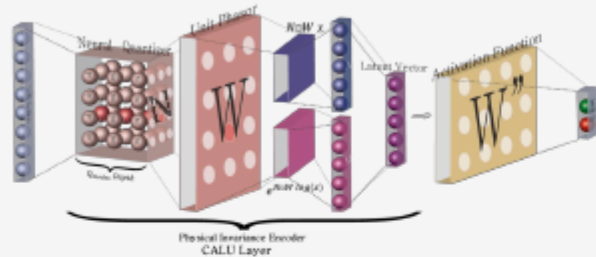
→ Re-discovering the Lorentz Invariance using CALU filters

→ Trained discriminative model becomes universal,
beyond the region of training sample

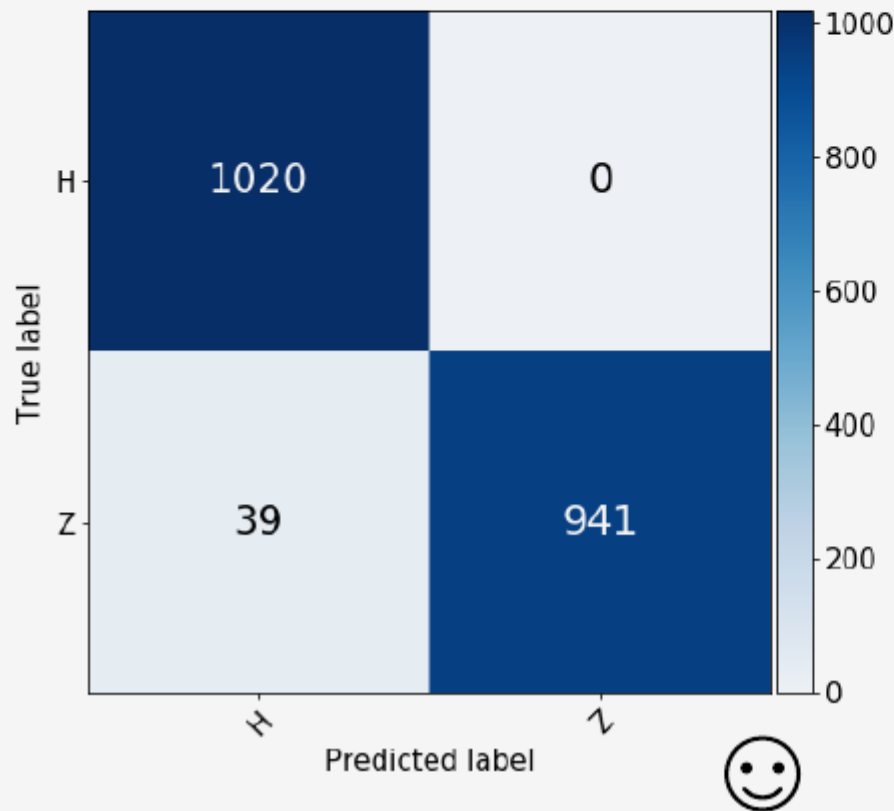
Experiments – 1. universality

Model Architecture	Data	Model structure
<p>①</p>  <p>8 200 200 200 2</p>	<p>** Train **</p> <p>$pp \rightarrow H/Z \rightarrow \mu\mu + jets$ $\sqrt{s} = 13 \text{ TeV}$ # of Data : 10,000</p>	<p>8:identity 200:ReLU 200:ReLU 200:ReLU 2:softmax</p>
 <p>8 100 3 50 2</p>	<p>Input = $(E_{\mu 1}, \vec{P}_{\mu 1}, E_{\mu 2}, \vec{P}_{\mu 2})$</p> <p>Output = (H, Z)</p> <p>** Test **</p> <p>$pp \rightarrow H/Z \rightarrow \mu\mu + jets$ $\sqrt{s} = 100 \text{ TeV}$ # of Data : 2,000</p>	<p>8:identity 100:CALUm 3:CALUa 50:ReLU 2:softmax</p>

Results of Classification – 1. universality



Acc = **98.1%**

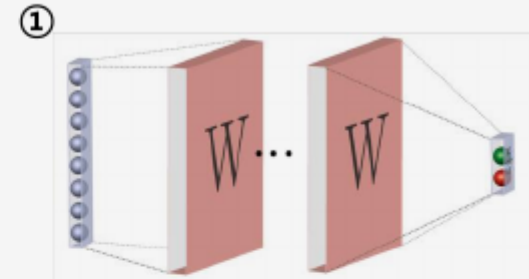


**** Train ****

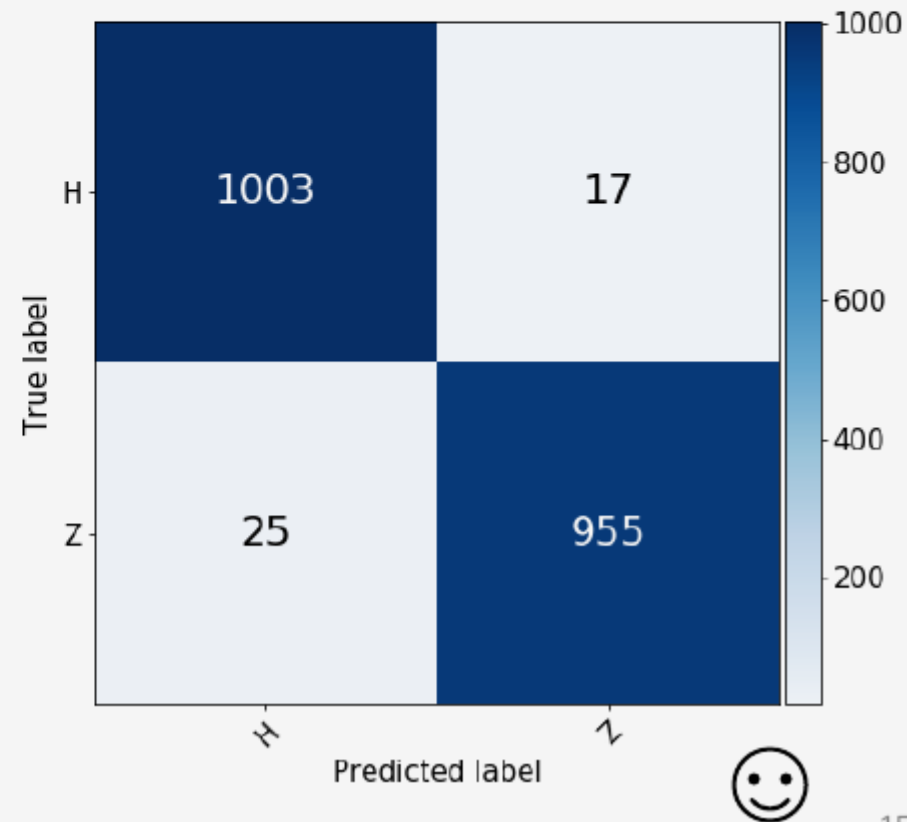
$pp \rightarrow H/Z \rightarrow \mu\mu + jets$

$\sqrt{s} = 13 \text{ TeV}$

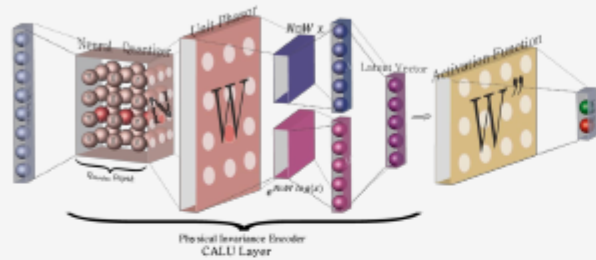
of Data : 2,000



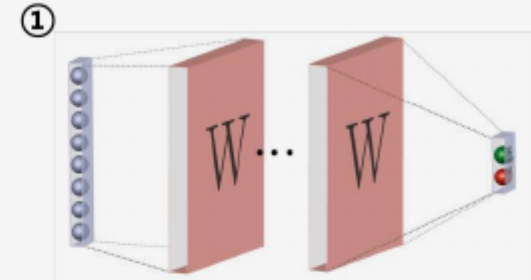
Acc = **97.9%**



Results of Classification – 1. universality



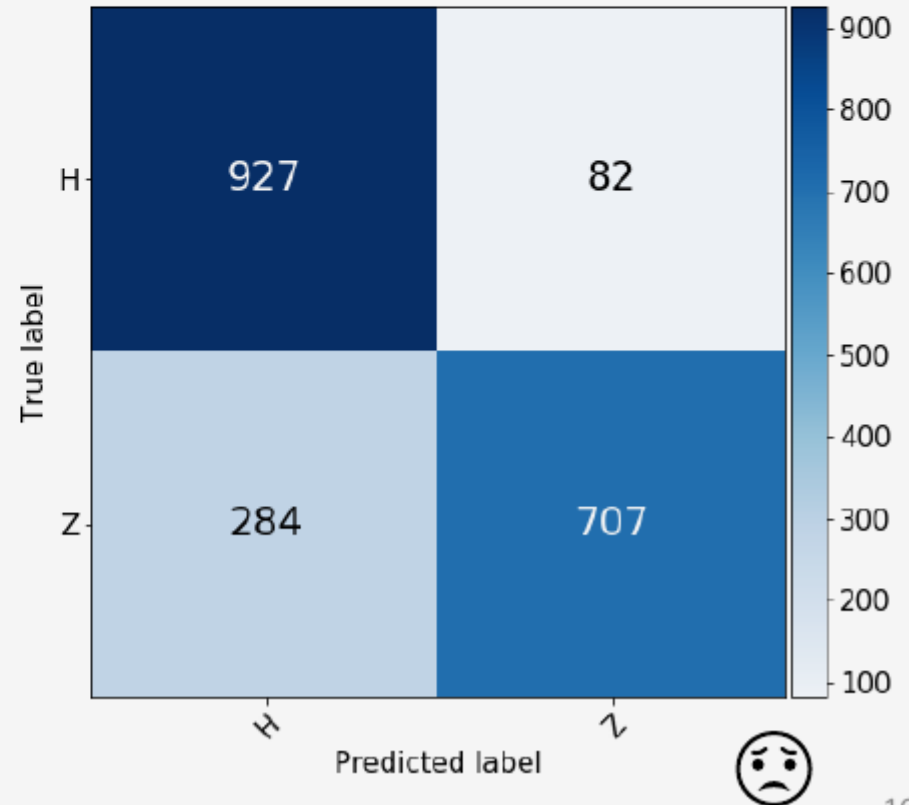
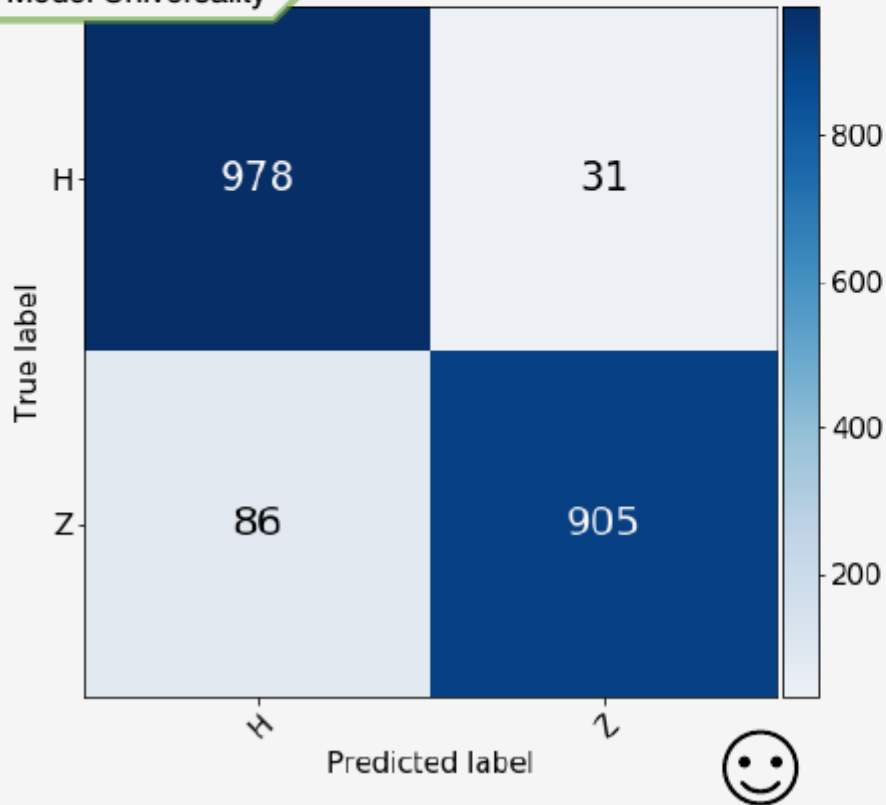
**** Test ****
 $pp \rightarrow H/Z \rightarrow \mu\mu + jets$
 $\sqrt{s}=100 \text{ TeV}$
 # of Data : 2,000



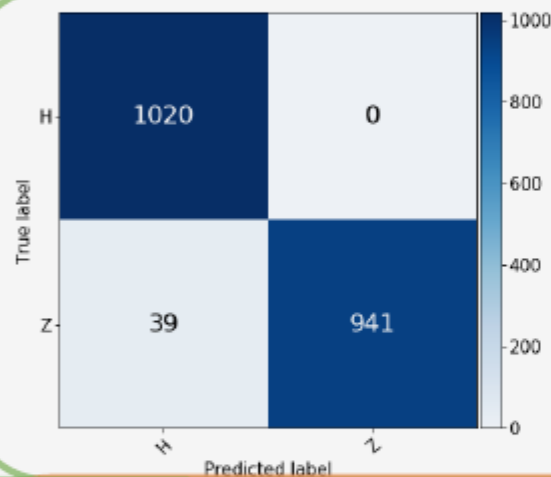
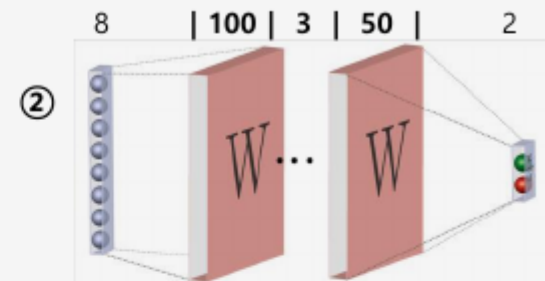
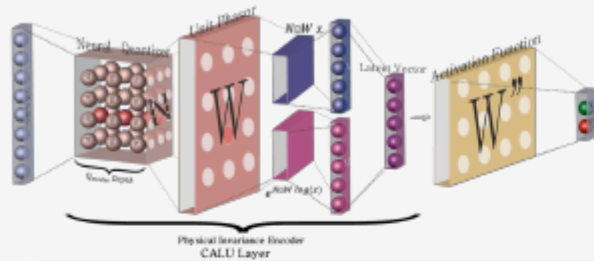
Acc = **97.8%**

Acc = **81.7%**

Model Universality



Results of Classification – 2. abstraction



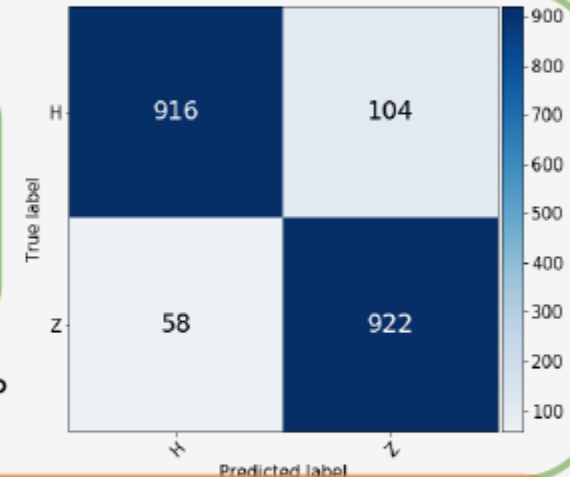
**** Train ****

$pp \rightarrow H/Z \rightarrow \mu\mu + jets$

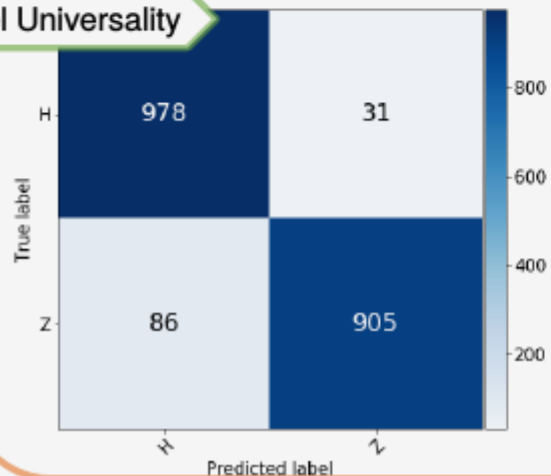
$\sqrt{s}=13 \text{ TeV}$

of Data : 2,000

Acc = **98.1%** Acc = **91.9%**



Model Universality



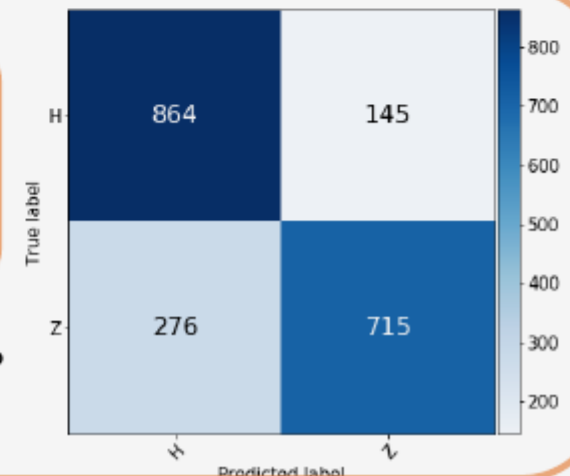
**** Test ****

$pp \rightarrow H/Z \rightarrow \mu\mu + jets$

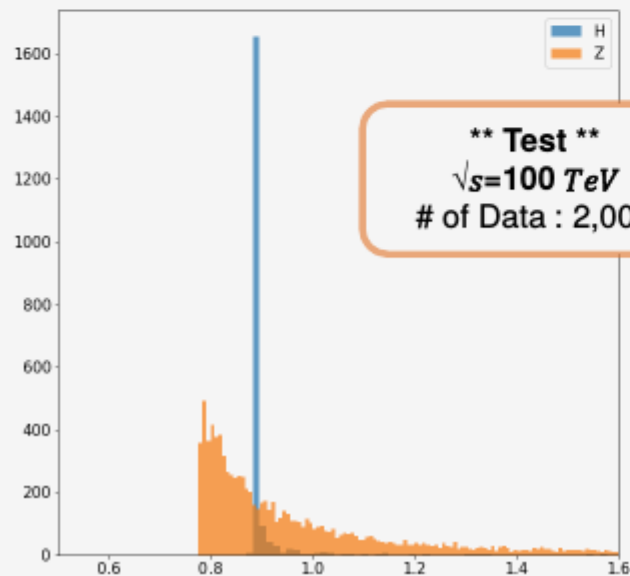
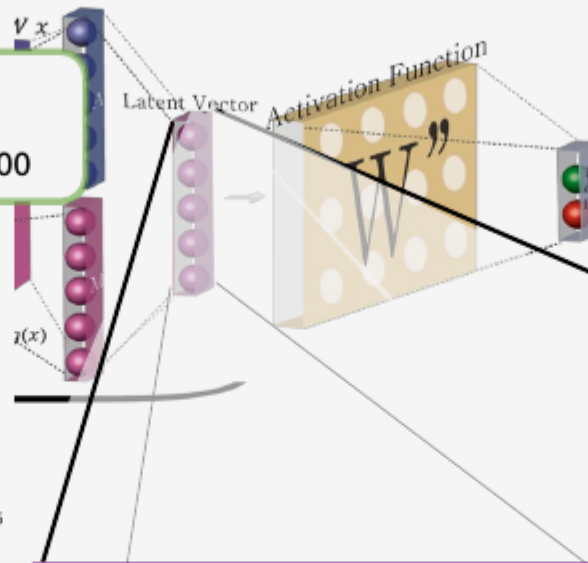
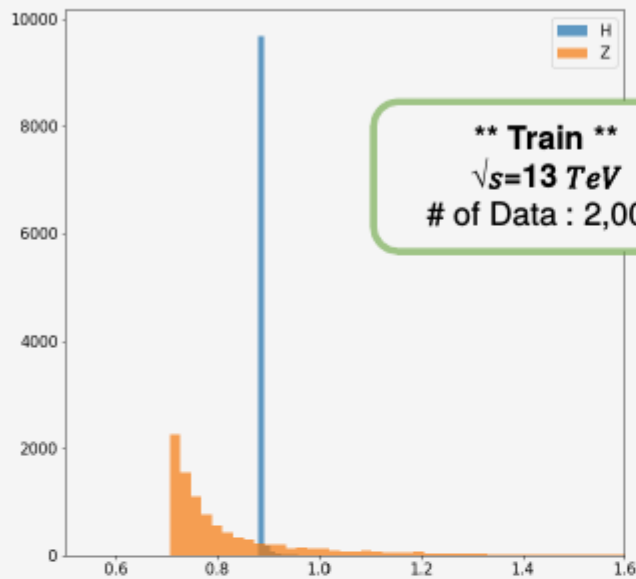
$\sqrt{s}=100 \text{ TeV}$

of Data : 2,000

Acc = **97.8%** Acc = **78.9%**



Discovery of the Lorentz invariance : CALU+DNN



1 Z1[0]

$$-0.01 E_1^{1.0} P_{y1}^{0.09} P_{z1}^{1.0} - 0.01 E_2^{1.0} P_{x2}^{0.12} P_{z2}^{1.0} + 0.01 P_{x1}^{1.0} P_{x2}^{1.0} - 0.02 P_{x1}^{1.0} - 0.01 P_{y2}^{1.0} + 2.0$$

1 Z2[0]

-1.0

1 Z3[0]

$$1.0 E_1^{1.0} E_2^{1.0} - 1.0 P_{x1}^{1.0} P_{x2}^{1.0} - 1.0 P_{y1}^{1.0} P_{y2}^{1.0} - 1.0 P_{z1}^{1.0} P_{z2}^{1.0}$$

Conclusion

- We designed a new NN architecture – **CALU with Neural Quantizer** which can learn the elementary arithmetic operations **in general complex-variable domain, trainable using back-propagation, in precision with universality.**
- **CALU nets have demonstrated its ability for extracting the exact physical invariance (Lorentz invariance) hidden in data, just under some classification pressure.**
- As an individual neuron (ex. caluon) gets more clear its own interpretation, dynamics between them becomes more and more important :
 - need for embedding symmetries and interactions among neurons
 - ex) Exclusive Quantization of caluon states
 - CALU representation of field theory, in general complex-domain
- Data-driven modeling and Principle/symmetry-based modeling are converging gradually...

ADVERTISEMENT

XAIENCE 2019

= e**X**plainable/crossing-over + **AI** + sci**ENCE**

- Workshop on AI applications **for science**, and **vice-versa**
- <https://www.xaience.cc>
- 2019 11.07 (Thu) – 08 (Fri), @ SNU
- We welcome your participation !

감사합니다