Discussion: Apply Learning to CANN

——Hebb Learning and Orthogonal Hebb Rule

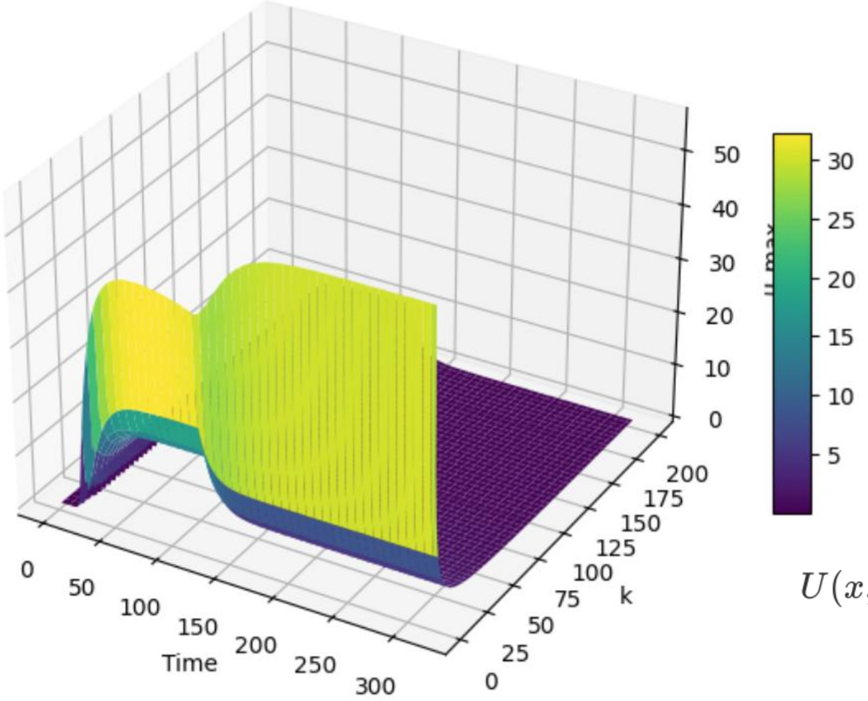
蔡锟瑾，毛川，贾仁和

2024.12.25

Finding's from CANN simulation: k and m

- Parameter k

The parameter k has a global inhibitory effect on the firing rate of neurons, regulating the overall activity level of the network.



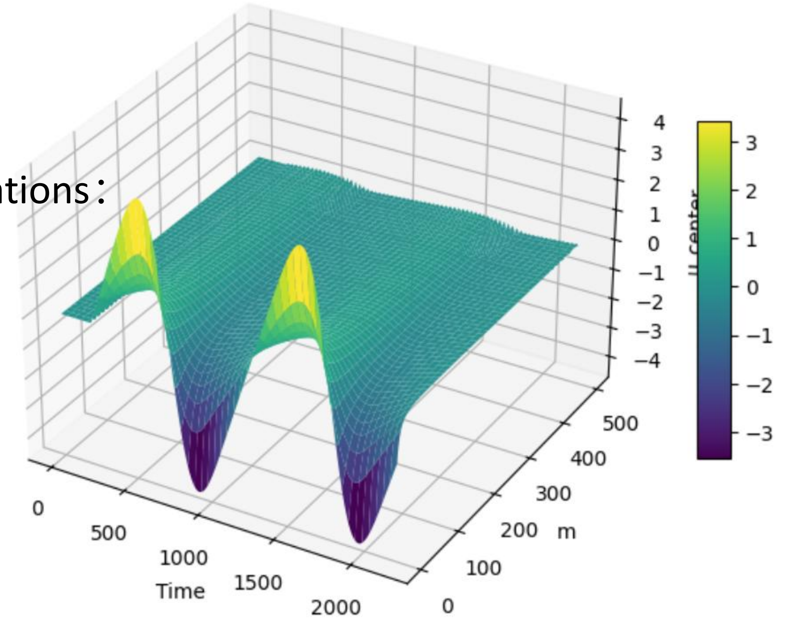
- Parameter m

When m is small, perform some rough calculations：

$$\tau_v \frac{dV(x,t)}{dt} = -V(x,t)$$

$$V(x,t) = A(x)e^{-\frac{t}{\tau_v}}$$

$$\tau \frac{dU(x,t)}{dt} = -U(x,t) + I^{rec}(x,t) + I^{ext}(x,t)$$

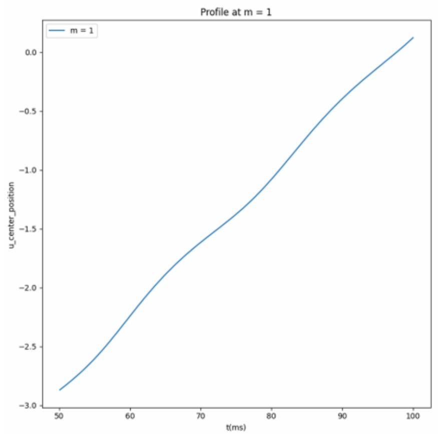


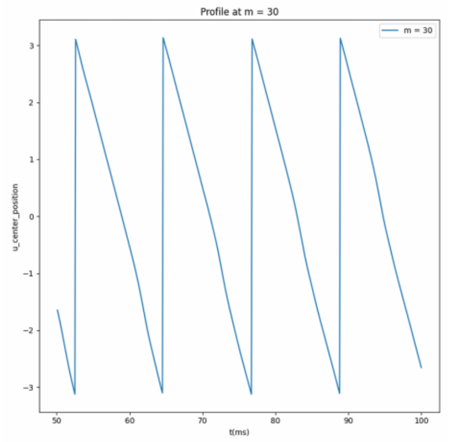
When m is big：

$$\tau_v \frac{dV(x,t)}{dt} = mU(x,t)$$

$$\tau \frac{d^2U(x,t)}{dt^2} = -mU(x,t)$$

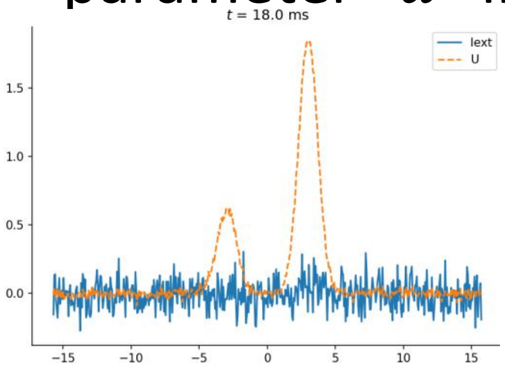
$$U(x,t) = A(x)\cos\left(\sqrt{\frac{m}{\tau}}t\right) + B(x)\sin\left(\sqrt{\frac{m}{\tau}}t\right)$$

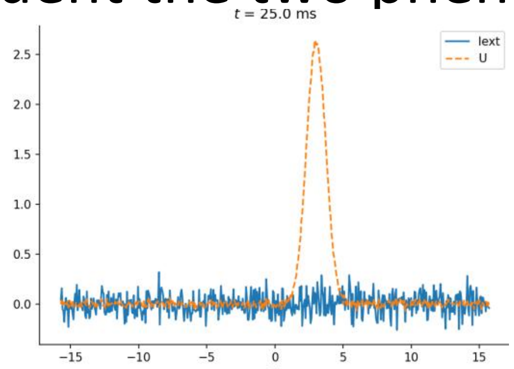




Investigation of Distance Correlation Among Multiple Stimuli

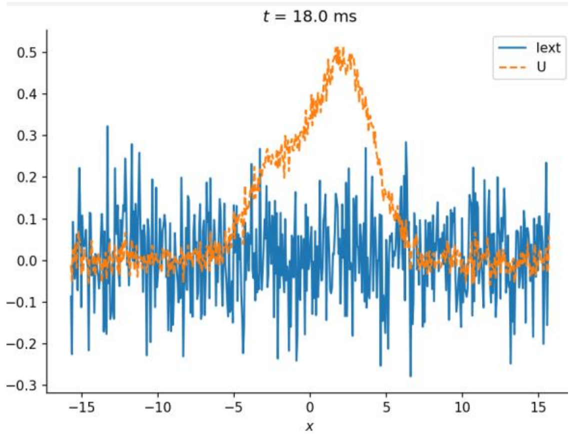
- If two difference input is not near, input at the same time.
- parameter “a” influent the two phenomenon



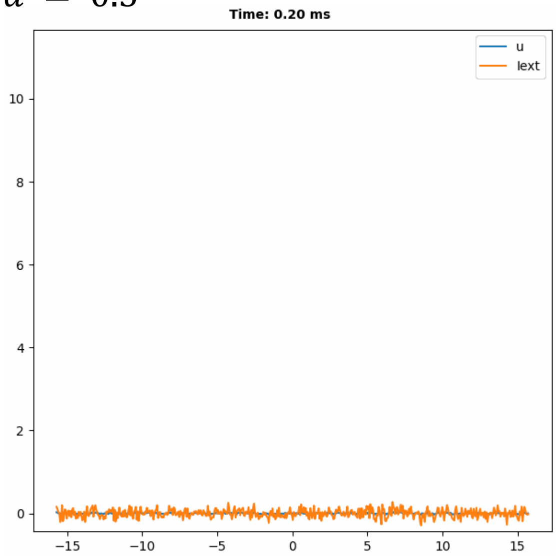


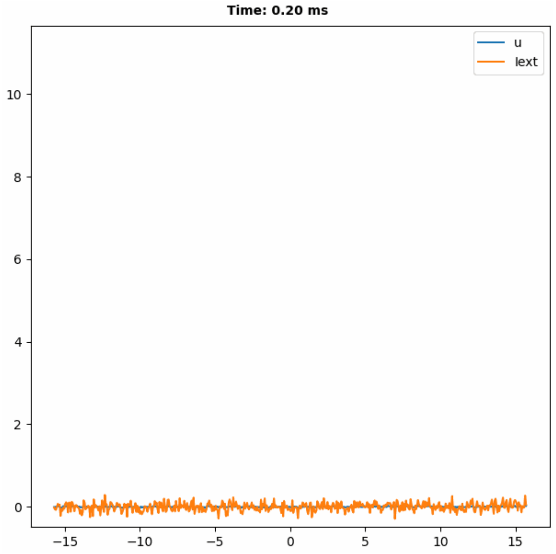
$$W_{ij} = A \cdot exp(-\frac{(x-x')^2}{2a})$$

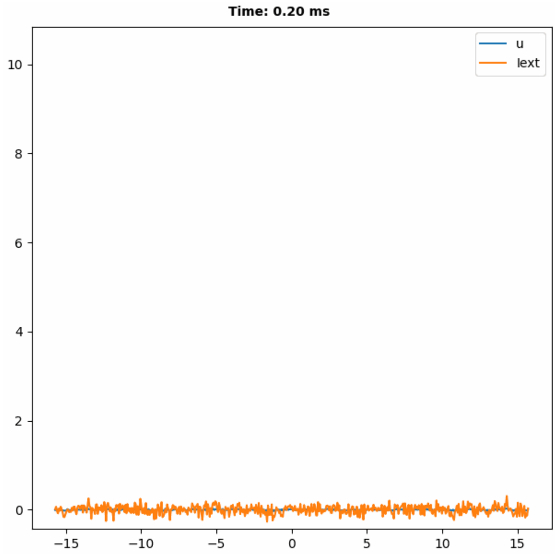
$a = 1.2$

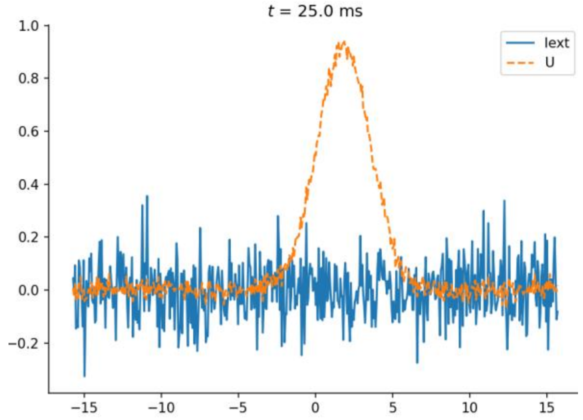


$a = 0.5$









Background

- If both neurons are activated at the same time, synaptic weights are enhanced.
- CANN can handle discrete inputs, smooth the input information, and reduce noise.
- By using patterns to train Hopfield Network, the matrix of synapse' connect strength can remember the features of patterns.
  - The stable states of Hopfield Network can encode a binary images.
- CANN can characterize the nature of continuous variation.
- Using some mathematical methods to improve traditional Hebb rule may improve the learning ability.
- Reference essay:
  - Xiaolong Zhou, eta. Learning a Continuous Attractor Neural Network from Real Images, ICONIP 2017
  - John J. Hopfield Neural Networks and Physical Systems with Emergent Collective Computational Abilities, PNAS

Apply Hebb Rule to 1D-CANN:

- Maybe directly apply Hebb rule to 1D-CANN which has encode distance continuity will destroy previous attribute of continuity and affects the ability of CANN to characterize continuous stimuli. But we think synapses will strengthen after continue activated, and the previous structure will change. Biologically, this mechanism may reflect the adaptive learning ability of neural networks in response to environmental changes.

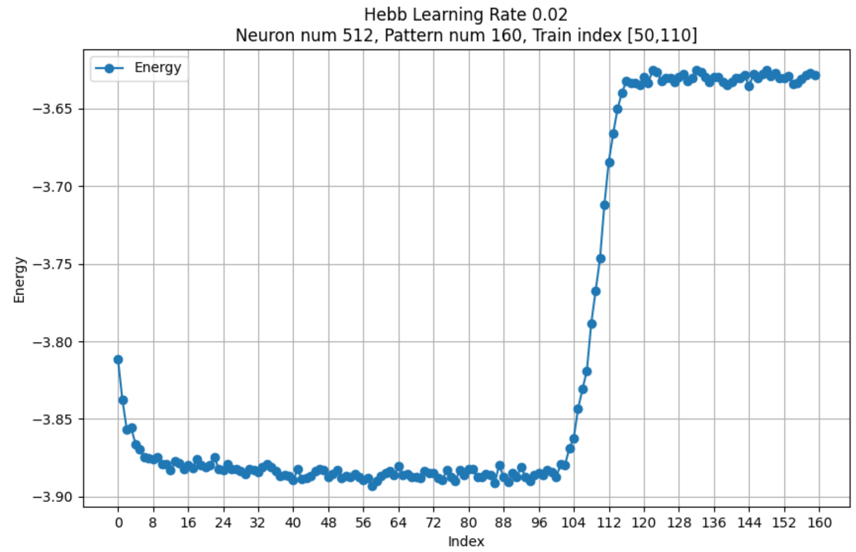
$$W_{ij} = A \cdot exp(-\frac{(x-x')^2}{2a})$$

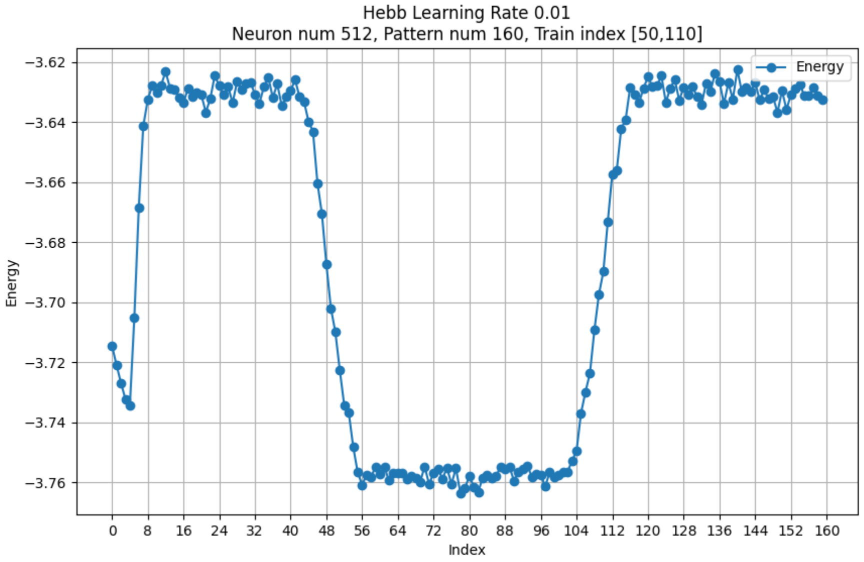
- We apply Hebb rule by input several times and record the voltage after remove input and network get stable.

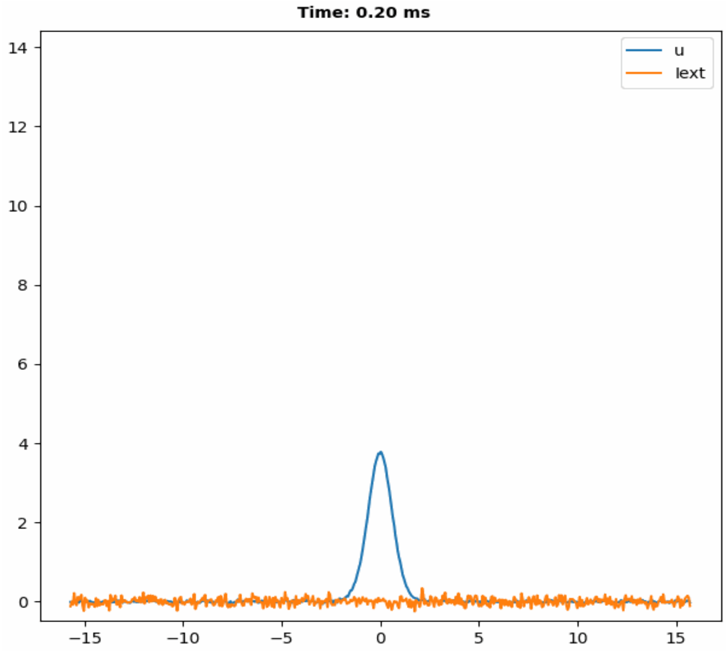
$$\Delta W(x,x') = \eta \cdot r(x) \cdot r(x')$$

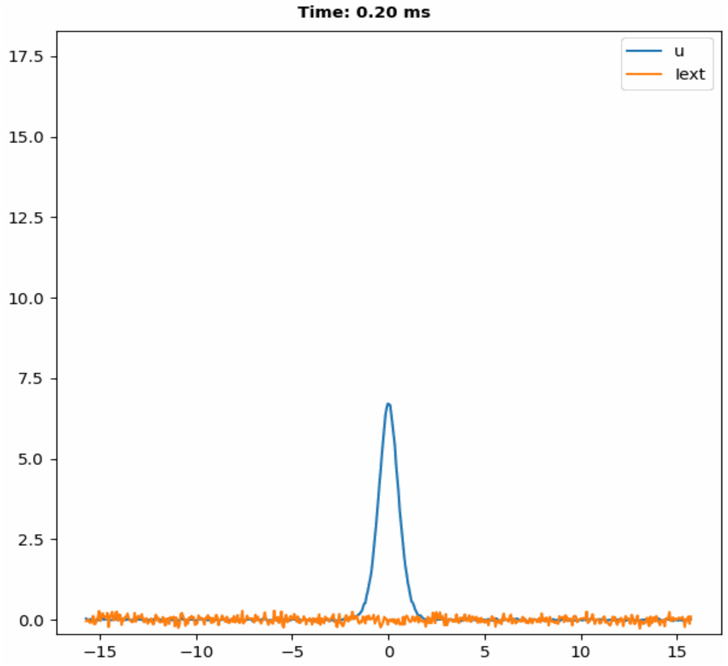
$$where \qquad r(x) = \frac{U^2(x)}{1 + k\int_x U^2(x)dx}$$

Hebb Rlue + 1D-CANN :learning result









Visual Image: Input to one place 8 time and change input to the nearby location.(previously learned 6 times)

The model can memory some features of the patterns if k is set appropriately. W can encode distance continuity as well as patterns' features.

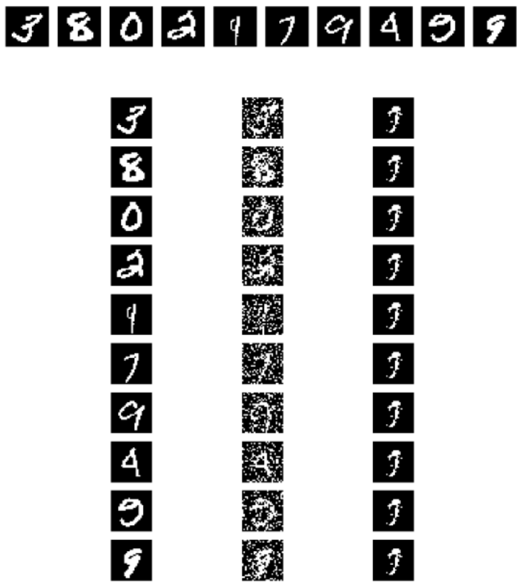
The way to calculate energy of the network:

$$E = -\frac{1}{2}V^T WV$$

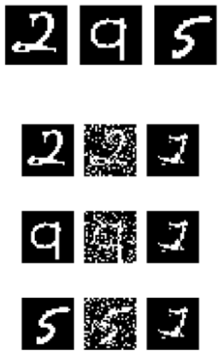
Apply Hebb Learning to Hopfield Network

- We also apply Hebb learning to a discrete network to test its ability of denoising.
- At the beginning, we calculate the W with patterns to be trained. We assume W has learned the input patterns' features.
- Task: denoise binary hand-written digits

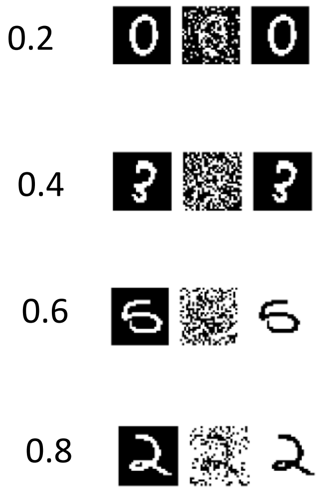
train 10 patterns



train 3 patterns



train 1 patterns (noise level)



$$w_{ij} = \frac{1}{p}\sum_{\mu=1}^{p} \eta_i^{\mu}\eta_j^{\mu} \qquad \forall i \neq j$$

$$S_i(t+1) = sign(\sum_{j=1}^{N} W_{ij}S_j(t))$$

$$E = -\frac{1}{2}S^T W S$$

Limitations of traditional Hebb learning

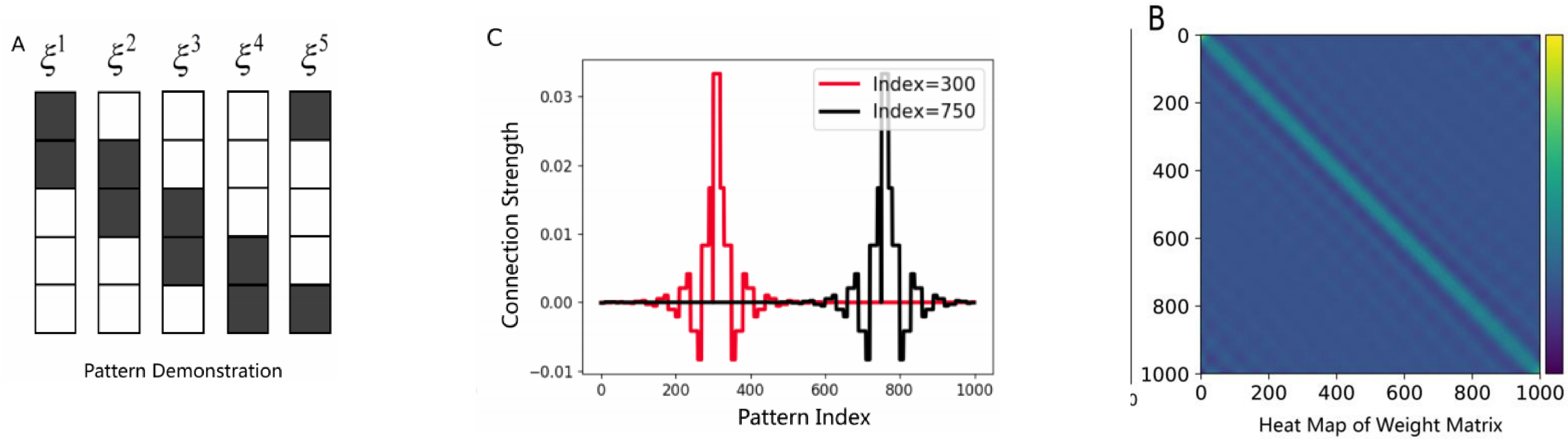
- Traditional Hebb Rule will "forget" former information which has been trained, if two input patterns are correlated. Because it can't distinguish difference.

- Modify pattern to be orthogonal, to strengthen the difference between the correlated pattern, which might improve memory ability.

- We want to apply an improved "Hebb learning method"——Orthogonal Hebb Rule

$$\eta^{p+1} = \xi^{p+1} - \sum_{\mu=1}^{p} \hat{\eta}^{\mu} \hat{\eta}^{\mu} \xi^{p+1},$$

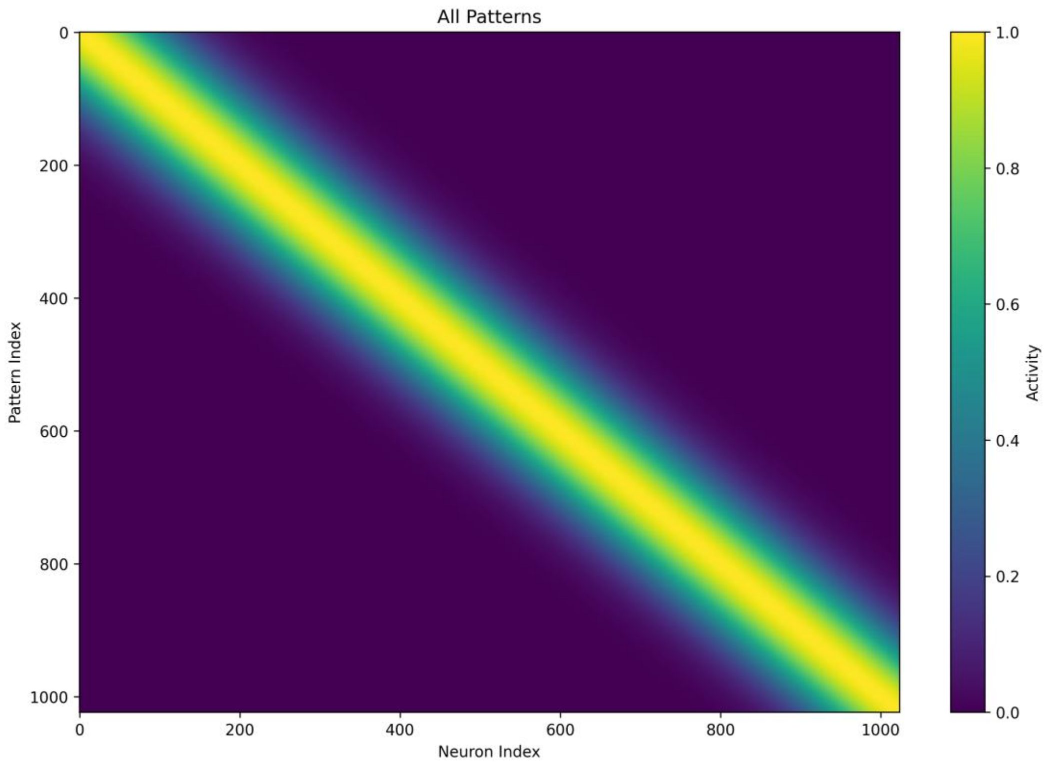
Orthogonal Hebb Rule——Continuous Attractor

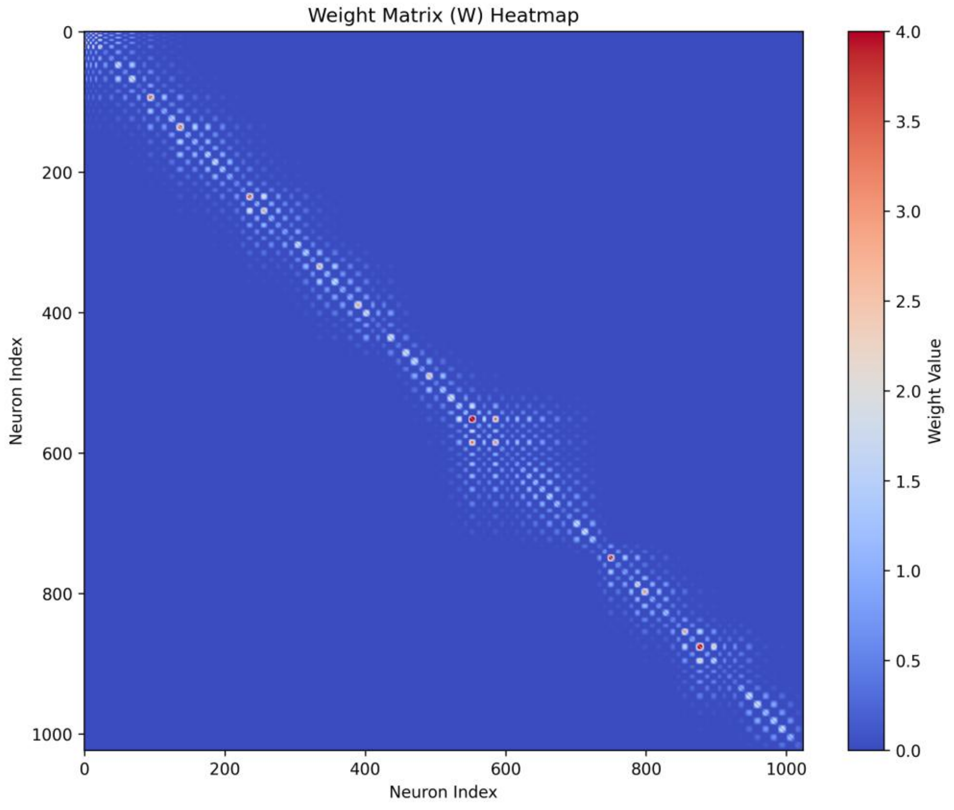


Zou, Xiaolong & Ji, Zilong & Liu, Xiao & Mi, Yuanyuan & Wong, K. Y. Michael & Wu, Si. (2017). Learning a Continuous Attractor Neural Network from Real Images. 622-631. 10.1007/978-3-319-70093-9_66.

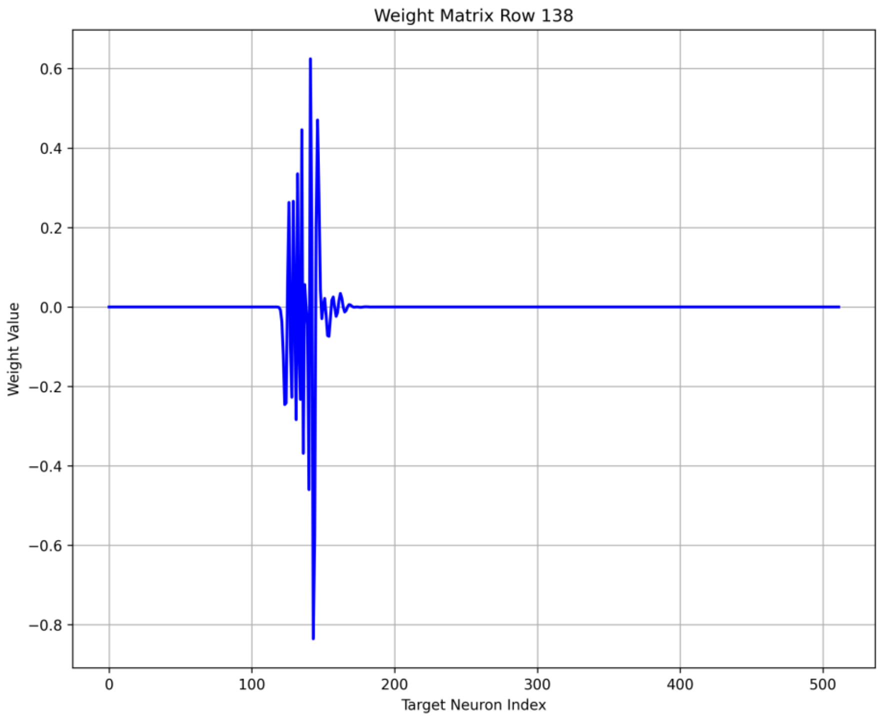
Orthogonal Hebb Rule——Continuous Attractor

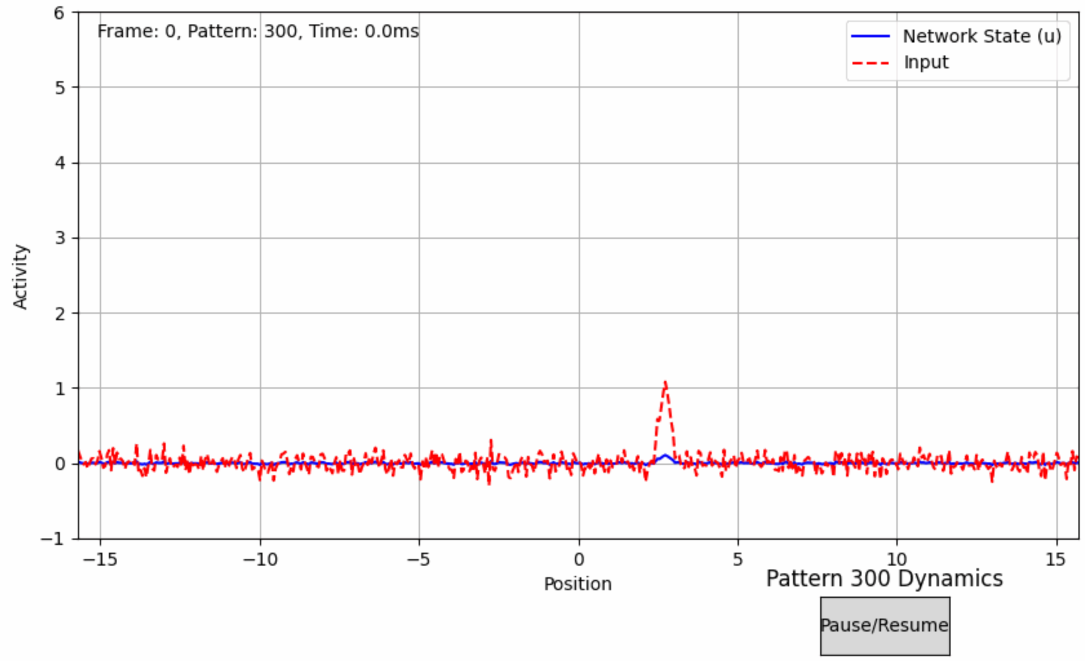
- We want to test the Orthogonal Hebb Rule in a continues space which is more like biological model.
- Task 1: Gaussian input



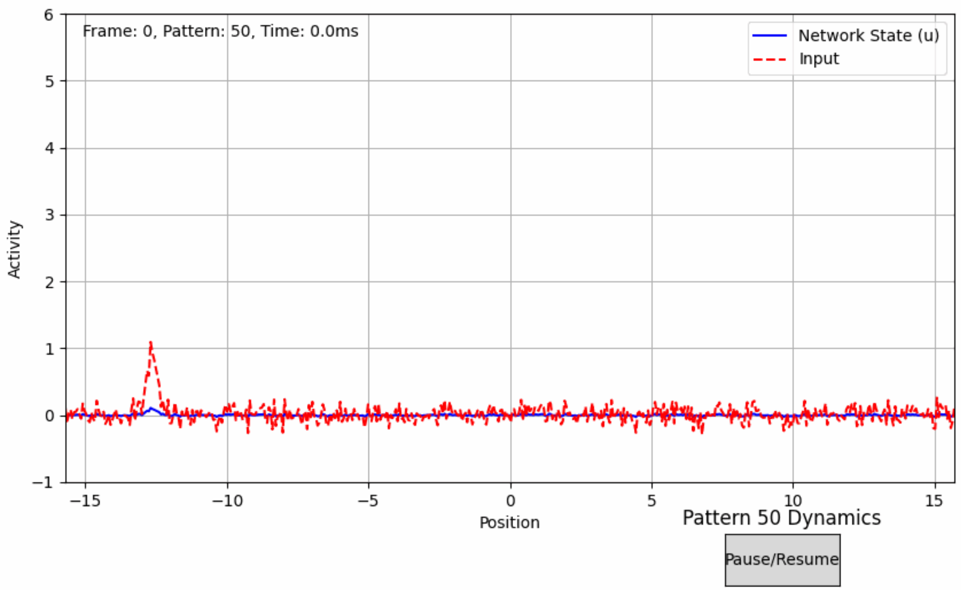


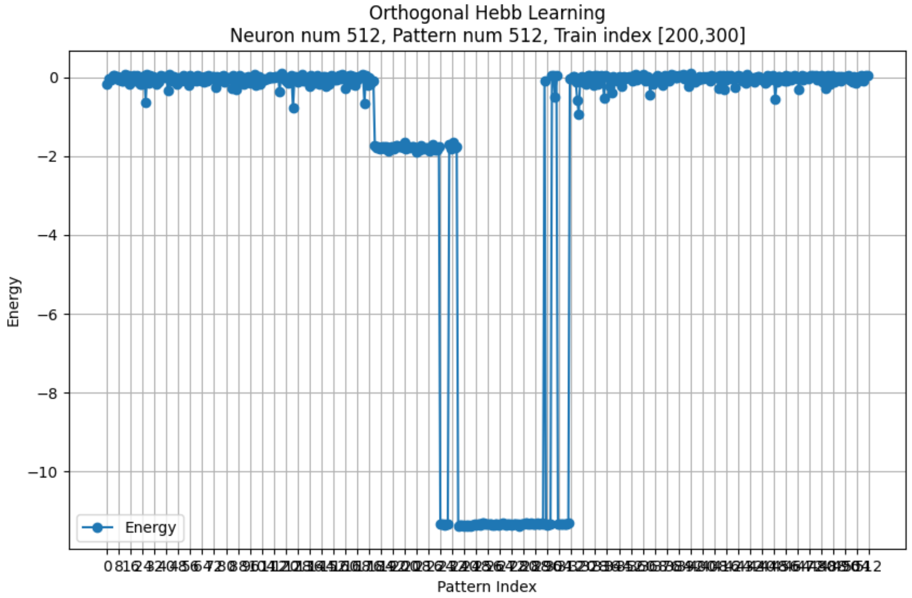
Orthogonal Hebb Rule——Continuous Attractor





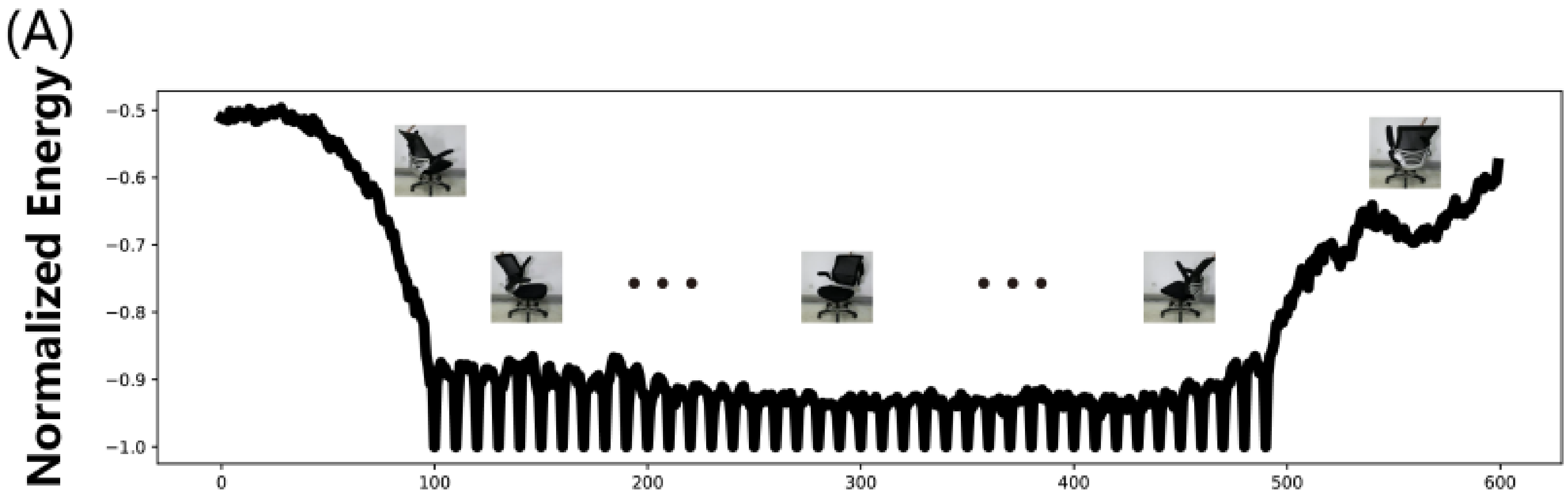
Orthogonal Hebb Rule——Continuous Attractor





Orthogonal Hebb Rule——Continuous Attractor

Rotating chair-character frature trained by VGG （Very deep convolutional networks for large-scale image recognition" (Simonyan & Zisserman, 2014) ）



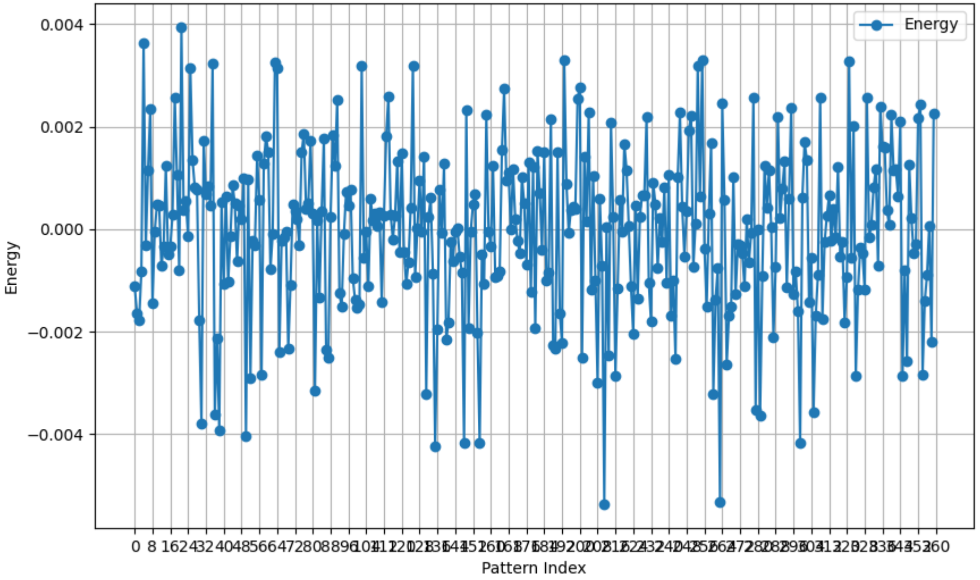
北京大學 PEKING UNIVERSITY

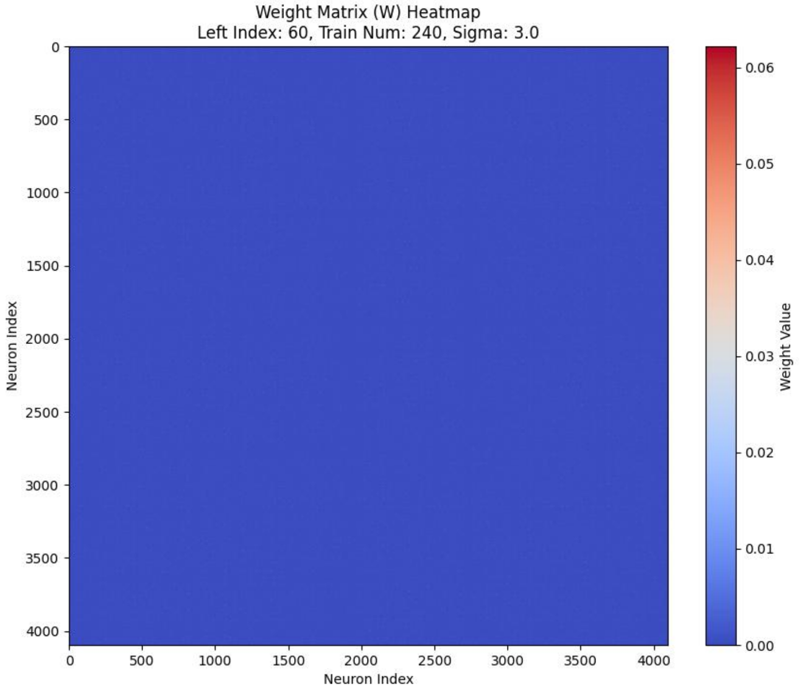
Orthogonal Hebb Rule——Continuous Attractor

Rotating chair-trained by VGG16

Orthogonal Hebb Rule——Continuous Attractor

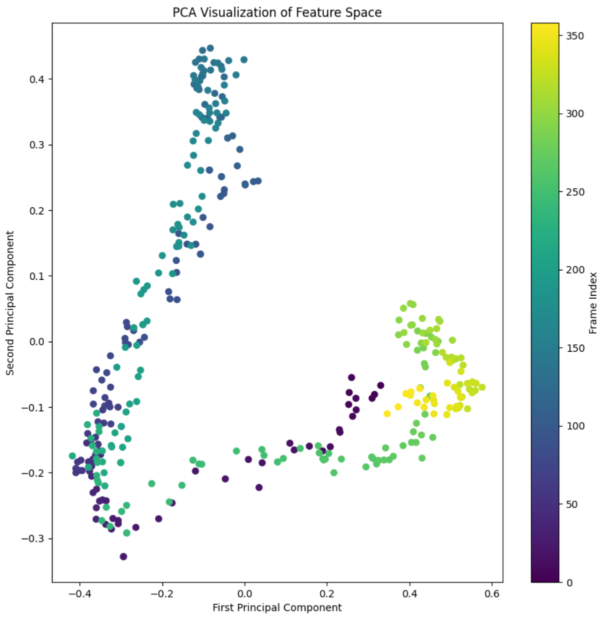
- Task 2: Use our orthogonal Hebb rule to encode continuous physical space
- Rotating chair

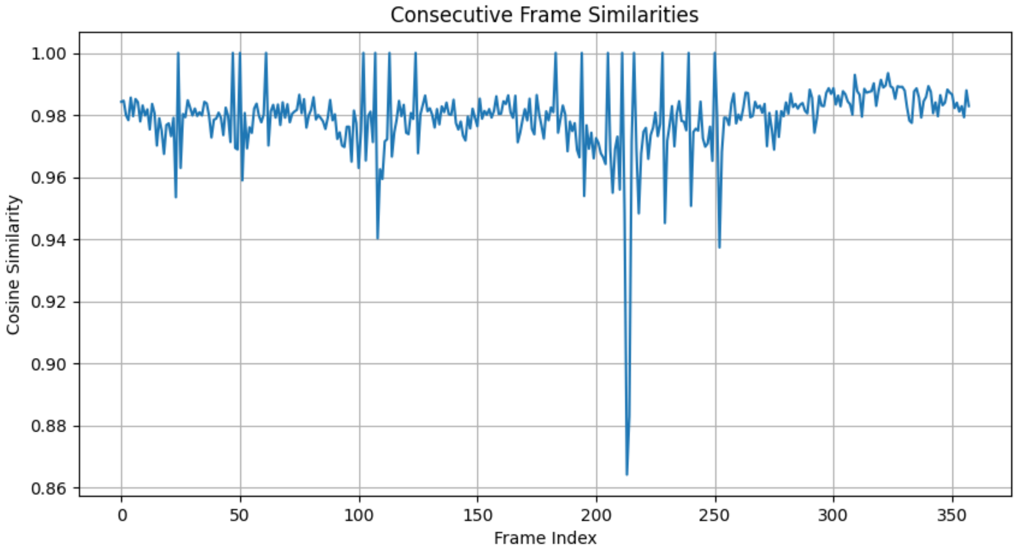


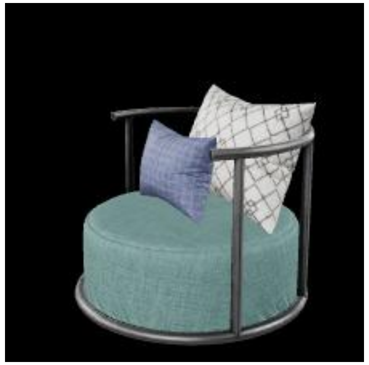


Orthogonal Hebb Rule——Continuous Attractor

- Task 2: Use our orthogonal Hebb rule to encode continuous physical space
- Rotating chair->change image and DNN

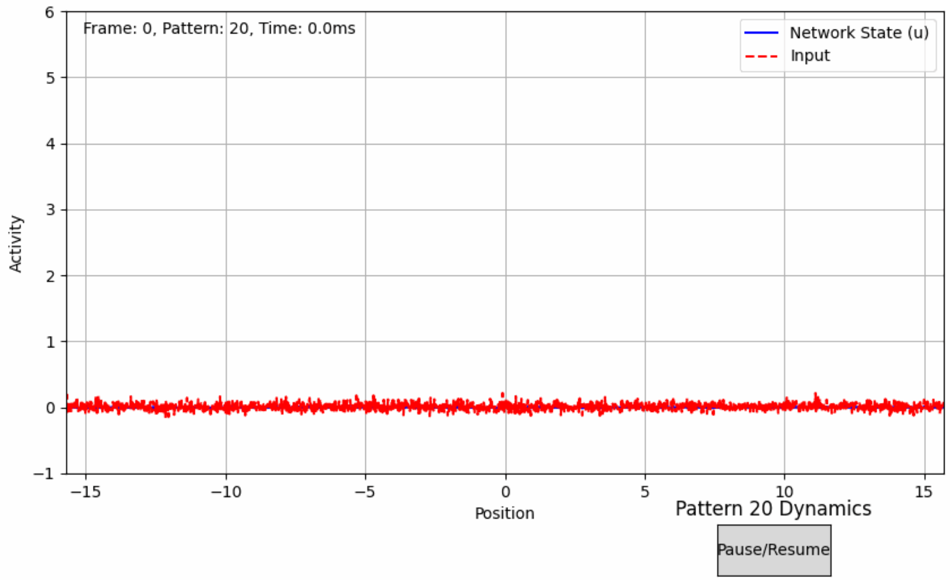


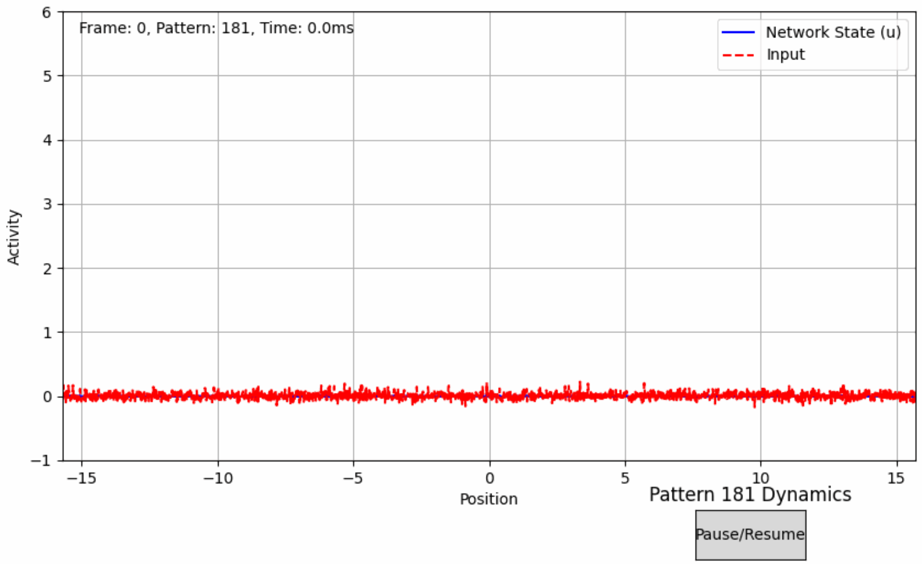




Orthogonal Hebb Rule——Continuous Attractor

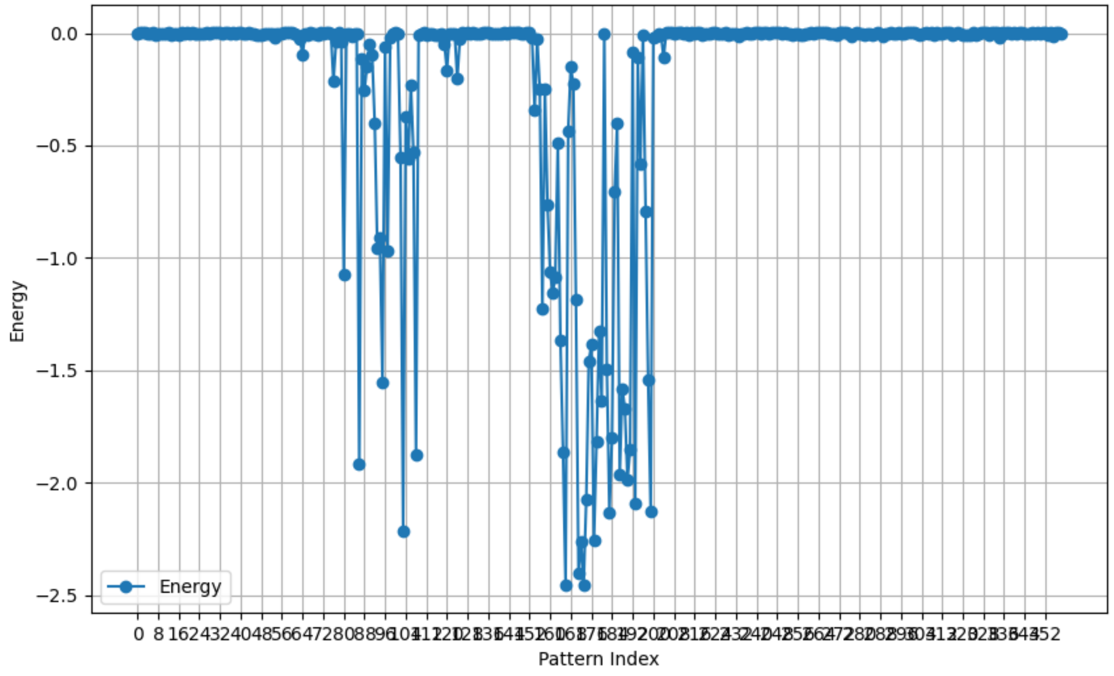
- Task 2: Use our orthogonal Hebb rule to encode continuous physical space
- Rotating chair trained by resnet50





Orthogonal Hebb Rule——Continuous Attractor

- Task 2: Use our orthogonal Hebb rule to encode continuous physical space
- Rotating chair trained by resnet50



Further Topic

- CANN may be capable of handling more complex image tasks.
- Attempt to use a biological model to vectorize images which can simulate retinal encoding.
- Encode high-dimensional information such as angle, distance, and brightness by coupling multiple CANNs.

Thanks!