Neutralized Synchronic and Diachronic Potentiality for Interpreting Multi-Layered Neural Networks

Ryotaro Kamimura
Tokai University

Outline

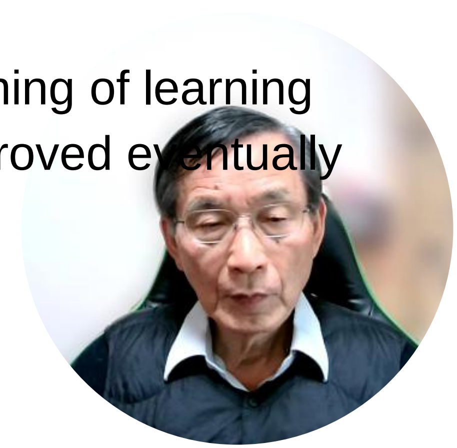
- **Simplification forces**
  - Strong tendency for simplification
  - The simplest form is realized by the prototype

- **Prototype detection by potentiality reduction**
  - The prototype detection is realized by simplifying network configurations by synchronic potentiality reduction

- **Neutralization or weakening potentiality reduction**
  - The synchronic potentiality is sometimes too strong in simplifying network configuration
  - Diachronic potentiality is introduced to neutralize or weaken the strong performance of synchronic potentiality reduction,

- **Application to artificial data set**
  - The prototype was detected in the beginning of learning
  - At the same time generalization was improved eventually by neutralization of two potentialities

- **Related and future works**
  - Implicit regularization



Simplification Forces

- **Simplifying network configuration**
  - Neural networks have strong forces to simplify network configurations as well as input patterns
  - Only by this simplification, neural networks can deal with seemingly complicated data sets
  - By dealing with only simpler portion of data set

- **Complexity against simplicity**
  - Complexity of networks and data sets seem to be contradictory to the principle of simplicity tendency

Simplification Anywhere

- **Simple components and procedures**
  - Neural networks are constructed in very simple way. Any components inside and learning procedures are simple
  - Only the collective behaviors of all simple components with simple learning procedures can realize seemingly complicated behaviors

- **Simplicity anywhere**
  - When we consider this fact, the simplicity seems to be more easily accepted
  - There are many examples to support this simplicity tendency
  - One of the typical examples is the implicit regularization

Simplification by Prototype

- **Prototype simplification hypothesis**
  - We here suppose that simplification is realized by prototype networks
- **Prototype**
  - The prototype networks are assumed to be the simplest ones within given network resources
- **Surface network production**
  - From it, we can generate surface complicated networks
- **Simplification and prototype**
  - Simplification forces can be observed by detecting the simplest prototype

Synchronic Potentiality Reduction for Prototype

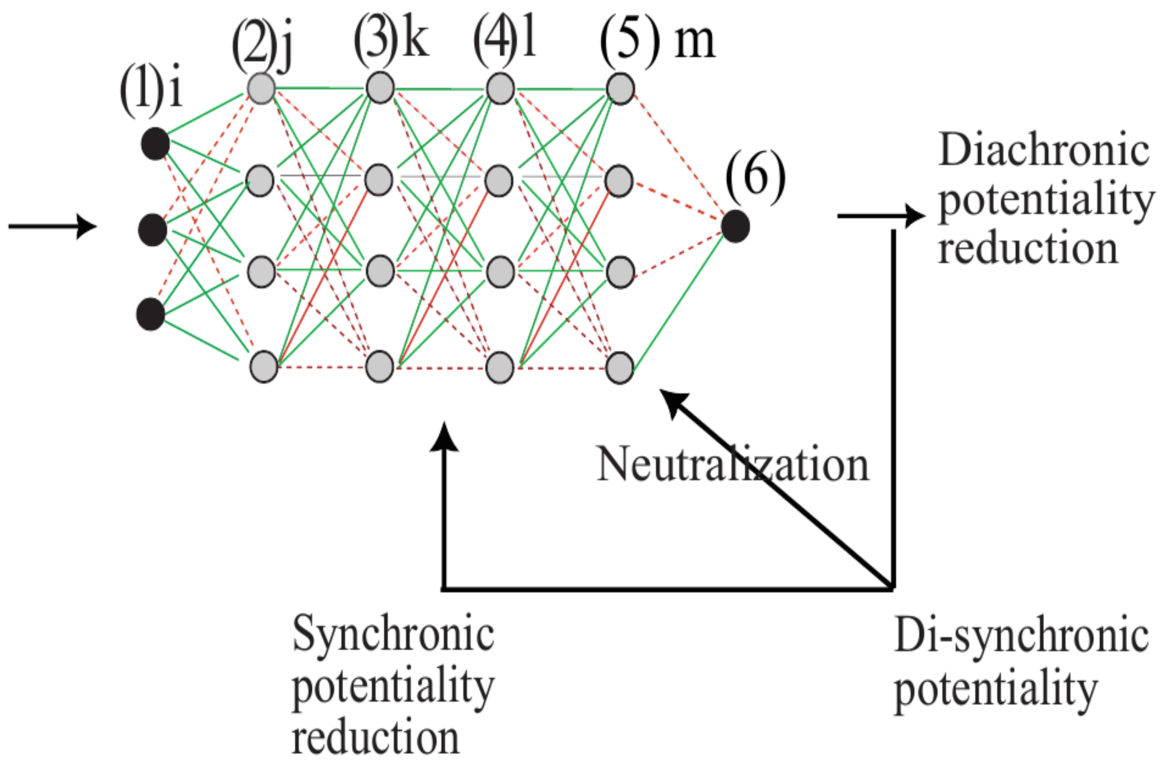
- **Prototype finding**
  - The protype is the simplest network, deeply hidden in the surface networks
  - We need to develop a method to make the prototype as overt as possible
  - To transform complex surface networks to the prototype, we need to eliminate unnecessary information
- **Synchronic potentiality for prototype**
  - Synchronic potentiality has been developed to reduce unnecessary information

Strong Synchronic Potentiality

- **Strong potentiality reduction**
  - The potentiality is a latent ability of component to be transformed into information necessary for structuring a network
  - The synchronic potentiality is sometimes too strong in reducing the potentiality
- **Neutralization or weakening**
  - For extracting important information, neutralization (weakening) of strong potentiality reduction is needed
  - For weakening learning, we try to use diachronic potentiality

Neutralization

- **Synchronic potentiality**
  - Strong reduction force
- **Time dependent neutralization**
  - Diachronic potentiality is used to weaken (neutralize) the strong potentiality reduction diachronically or time-dependently



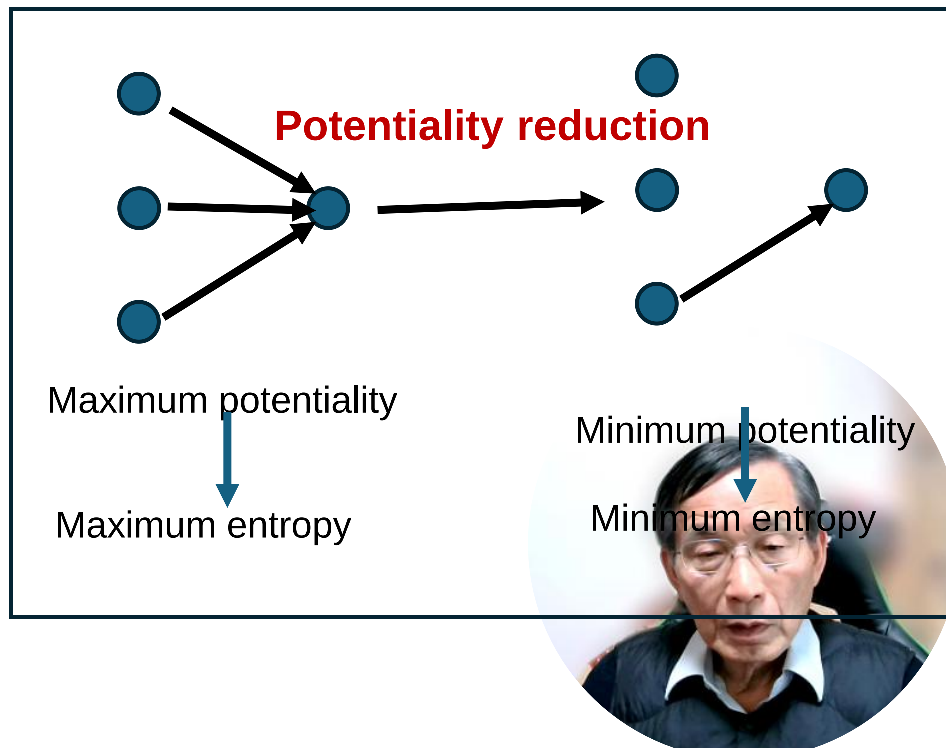
Synchoronic Potentiality Reduction

- **synchronic Potentiality**
  - Individual potentiality
    - The individual potentiality is the absolute connection weight
  - Relative potentiality
    - Individual potentiality should be divided by its maximum value
  - Potentiality
    - Potentiality is the sum of all relative individual potentialities

Individual potentiality
$$u_i^{(1,6)} = \left| w_i^{(1,6)} \right|$$

Relative potentiality
$$g_i^{(1,6)} = \frac{u_i^{(1,6)}}{\max_{i'} u_{i'}^{(1,6)}}$$

Potentiality
$$G^{(1,6)} = \sum_i \frac{u_i^{(1,6)}}{\max_{i'} u_{i'}^{(1,6)}}$$



- **Relation to entropy**
  - The potentiality is a simplified entropy function without the logarithmic function
  - The potentiality can be easily computed
  - This implies that the potentiality is very strong for simplification

Diachronic Potentiality

- **Synchronic potentiality**
  - Potentiality reduction is performed in a state, fixed at a learning time t

$$v_t = t$$

Individual time-dependent potentiality

- **Diachronic and time-dependent potentiality**
  - Diachronic potentiality is time-dependent
  - Diachronic potentiality is controlled with the parameter beta
  - This potentiality decreases as the parameter beta increases

$$h_t = \gamma_{dch} \left[\frac{v_t}{\max_{t'} v_{t'}} \right]^{\beta_{dch}}$$

Individual relative diachronic potentiality

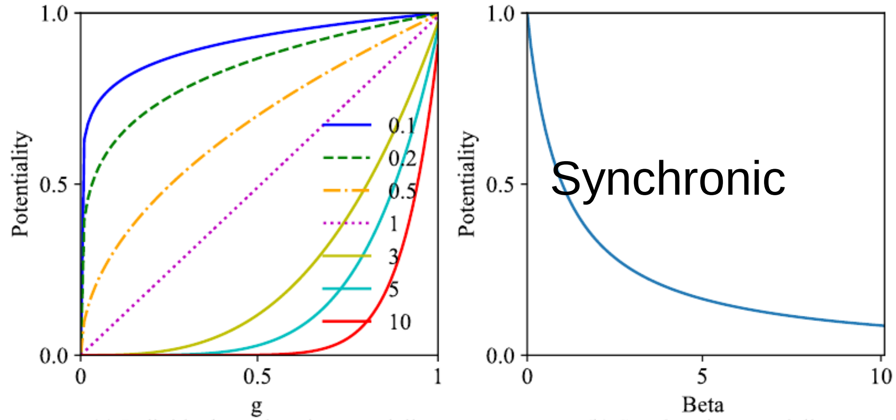
- **Neutralizing synchronic potentiality**
  - Synchronic potentiality reduction is neutralized by diachronic potentiality time-dependently

Neutralization

$$\text{neutral}_{jk}^{(n,n+1)}(t) = \gamma_{syn} \left[\frac{u_{jk}^{(n,n+1)}}{\max_{j'k'} u_{j'k'}^{(n,n+1)}} \right]^{h_t}$$

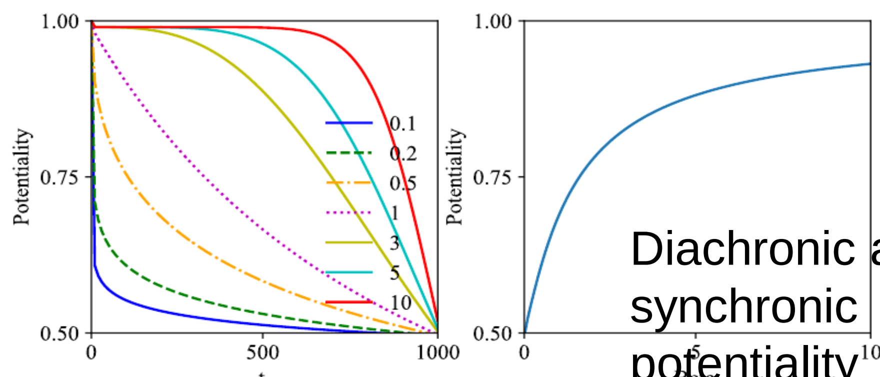
Neutralization

- **Synchronic potentiality**
  - Decreasing when the parameter beta increases
- **Diachronic potentiality**
  - Increasing when the beta increases
- **Neutralization**
  - aiming to increase synchronic potentiality implicitly by the diachronic potentiality increasing

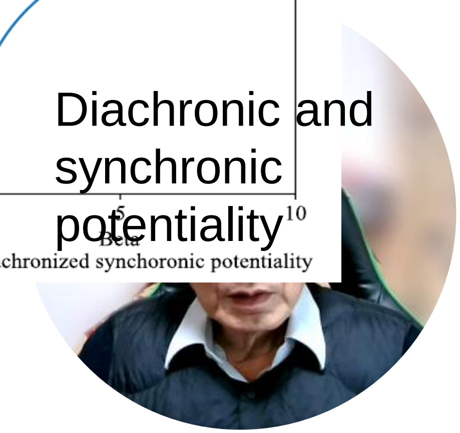


(a) Individual synchronic potentiality (b) Synchronic potentiality

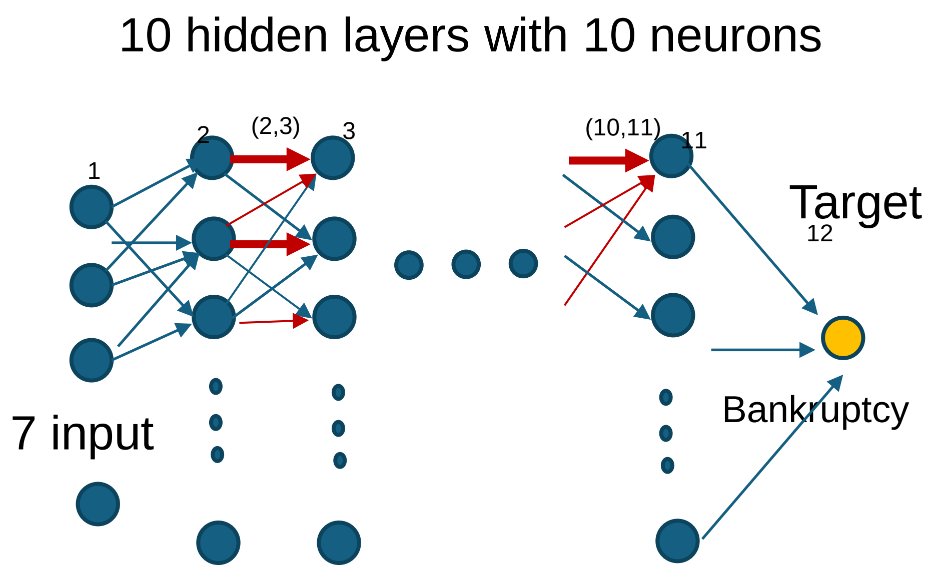
Neutralization



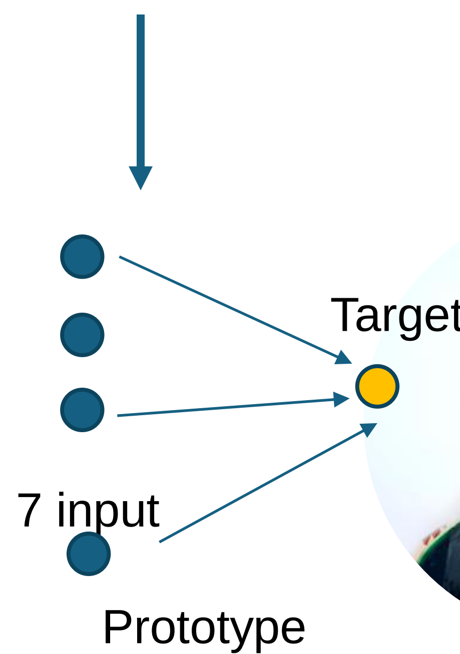
(a) Individual diachronized synchronic potentiality (b) Diachronized synchoronic potentiality



Application to Artificial Data Set

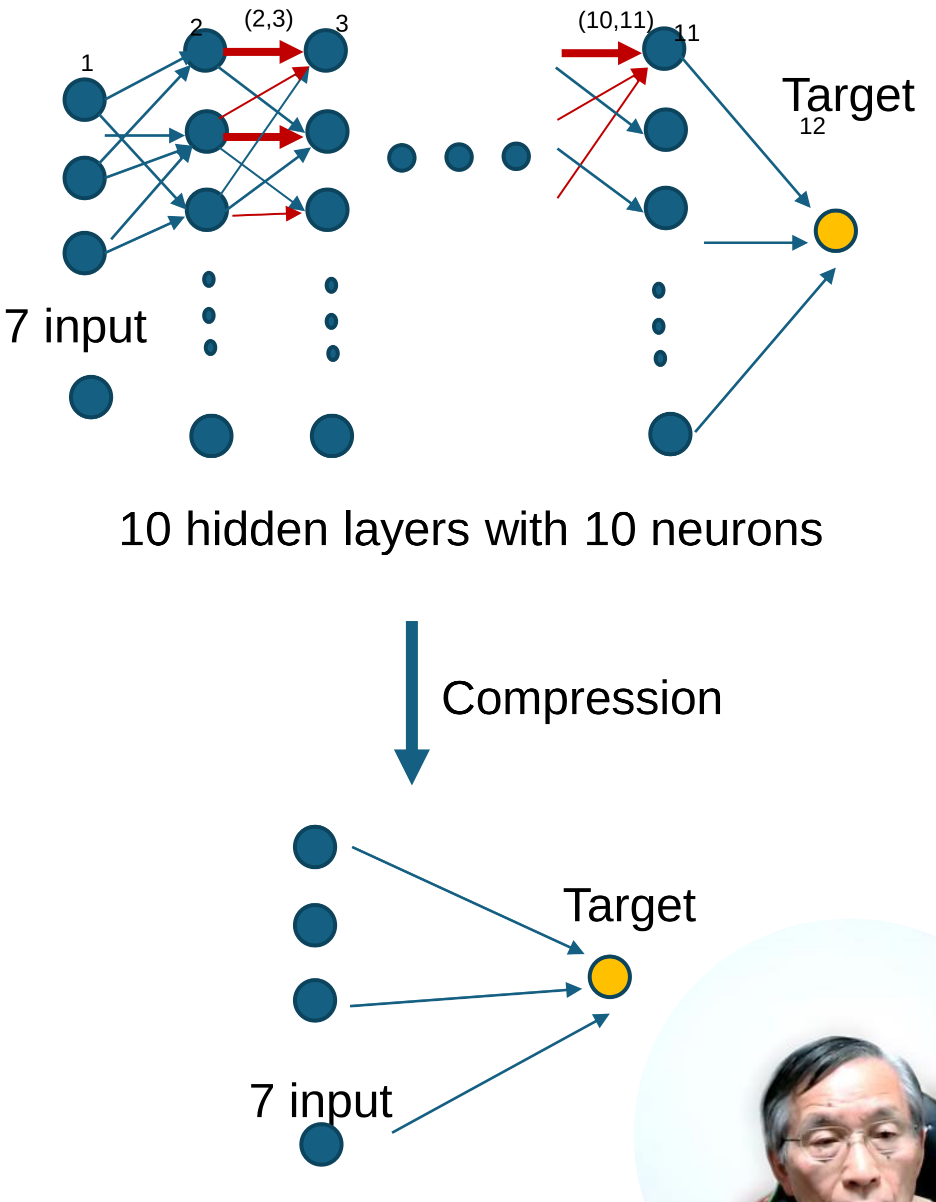


- **Interpretation**
  - The main objective is to interpret how a network responds to an input
  - The network is forced to be the simplest one and close to the prototype for interpretation
  - Because the prototype is the simplest one, the interpretation is much easier

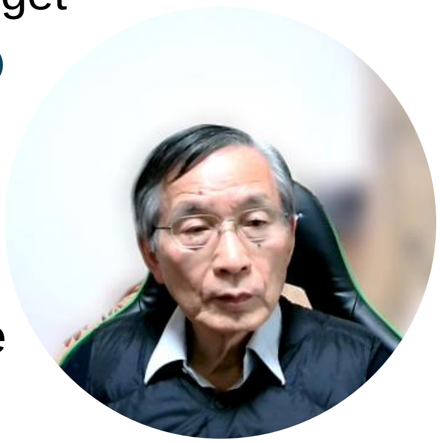


Estimating Prototypes

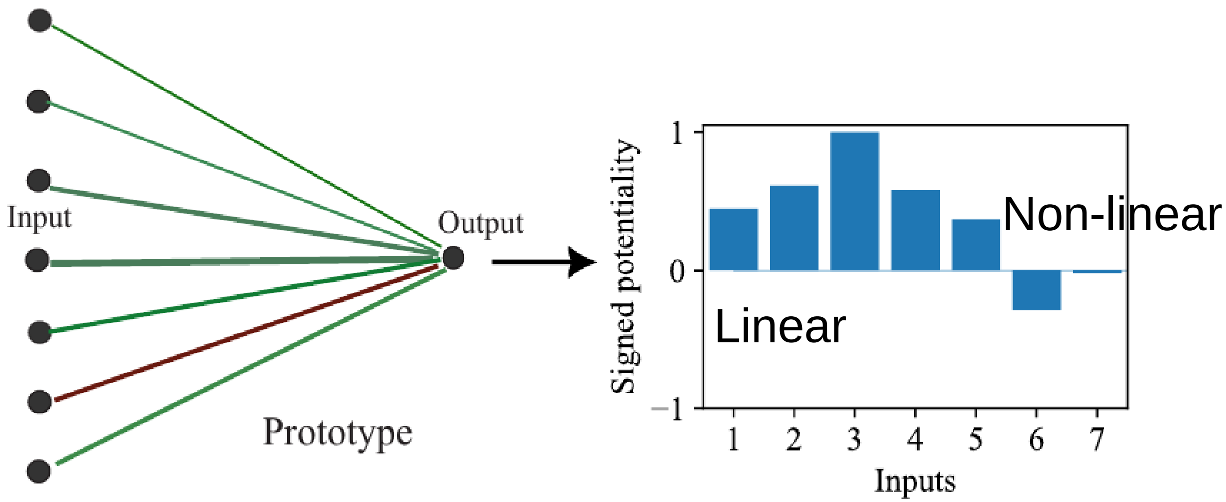
- **Compression**
  - The prototype should be estimated by compressing actual multi-layered neural networks
  - For example, a multi-layered neural network is compressed into a network without hidden layers by supposing that all activation functions are linear
  - This simplest network is used to estimate the prototype
- **Comparison**
  - We try to compare the supposed prototype with the corresponding estimated prototype



Estimated prototype

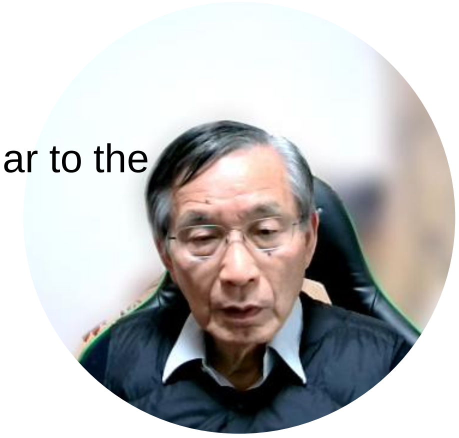


Data Description and Supposed Prototype



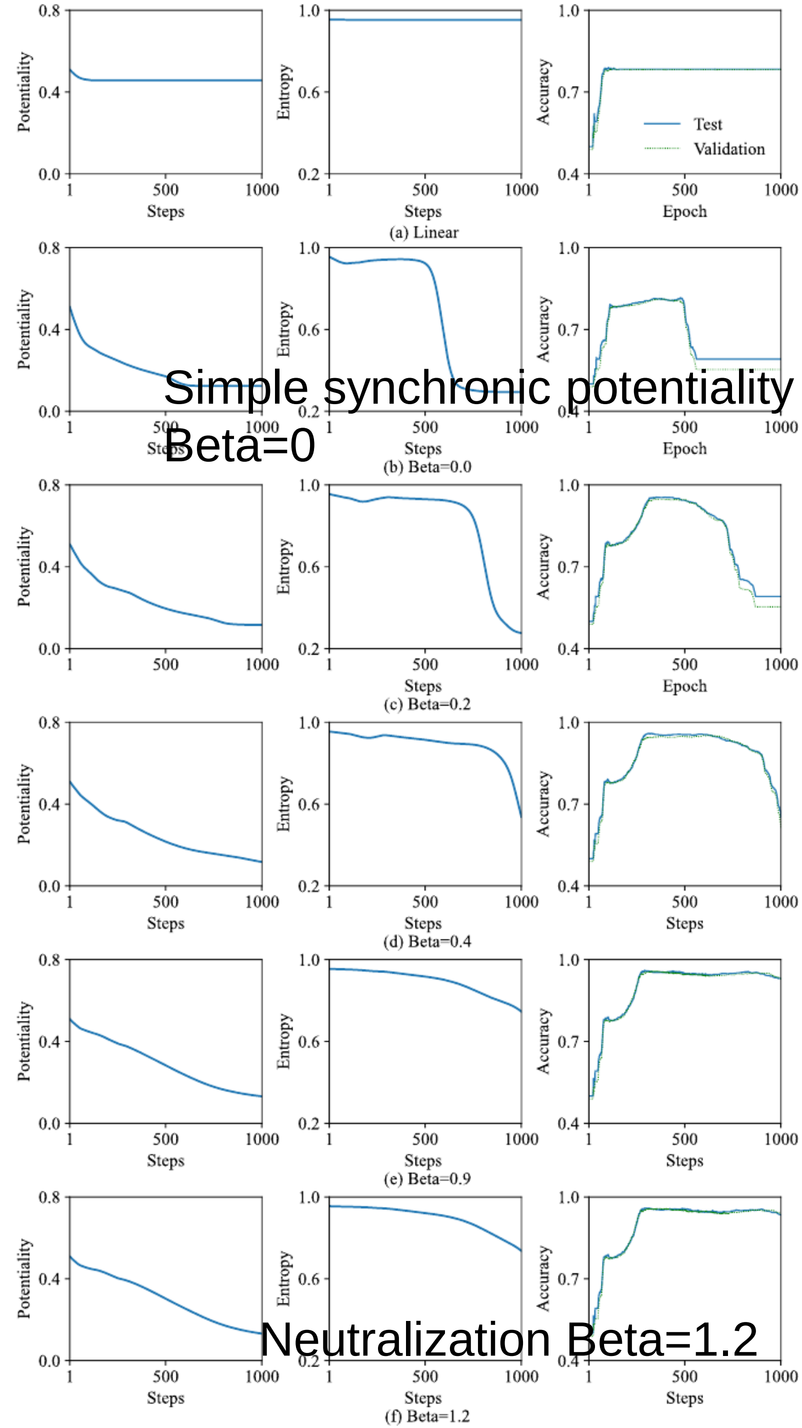
- **Data description**
  - An artificial data set was created with linear and non-linear inputs
  - Inputs from No.5, 6, 7 were created nonlinearly to the targets
  - The supposed prototype was created by taking the correlation coefficients between those inputs and targets

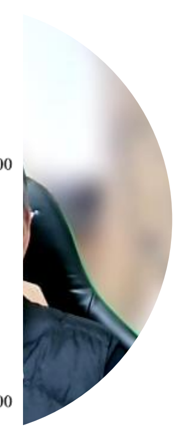
- **Objectives**
  - We tried to examine whether the estimated prototype was similar to the supposed prototype
  - We tried to examine how the prototype finding was related to generalization performance
  - We tried to interpret how the prototype responds to inputs



Potentiality Neutralization

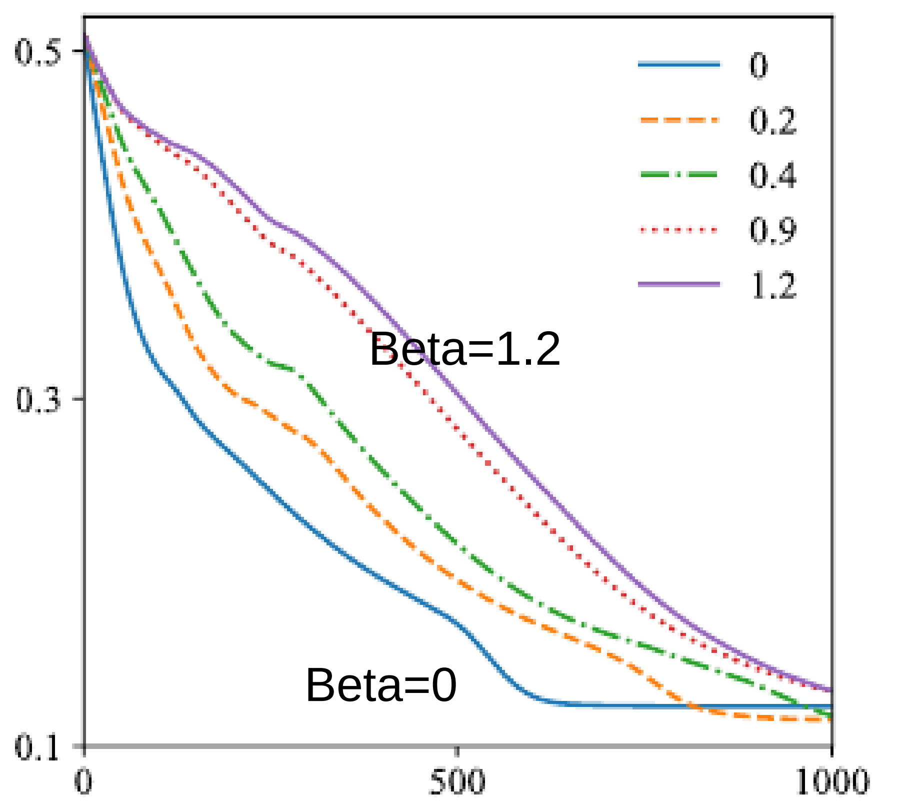
- **Simple synchronic potentiality**
  - When the simple synchronic potentiality was used (beta=0)
  - The potentiality decreased and the entropy decreased rapidly
  - Modest generalization
- **Neutralization**
  - As the parameter beta increased to 1.2, diachronic potentiality becomes effective
  - Synchronic potentiality reduction became weaker by using diachronic potentiality
  - Entropy reduction became also weaker
- **Improved generalization**
  - This weakening was related to improved generalization

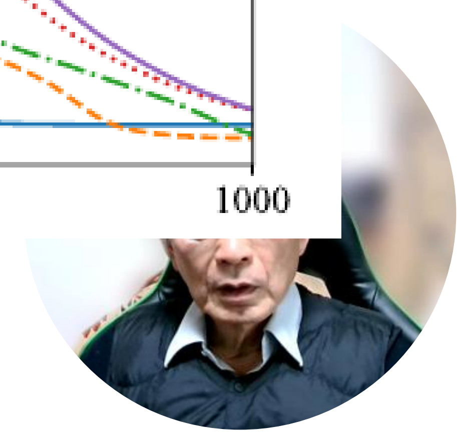




Effect of Neutralization

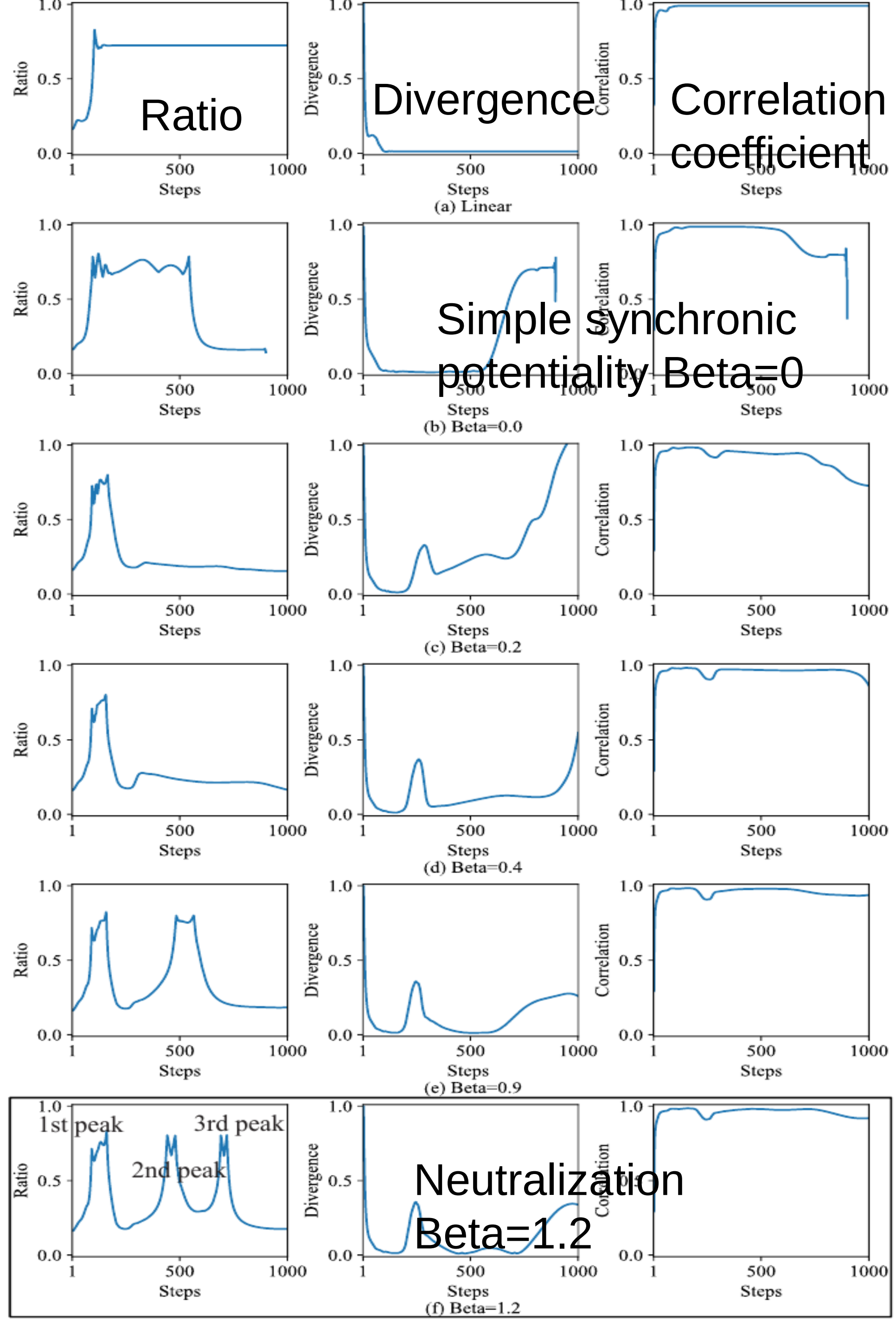
- **Neutralization**
  - When the parameter increased from 0 to 1.2, the force of neutralization becomes stronger
  - Synchronic potentiality became higher, when the parameter increased from 0 to 1.2
  - Weakening synchronic potentiality reduction could be realized by neutralization



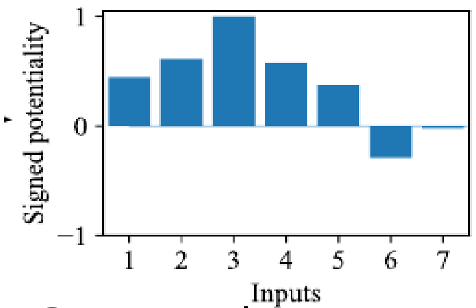


Ratio Potentiality

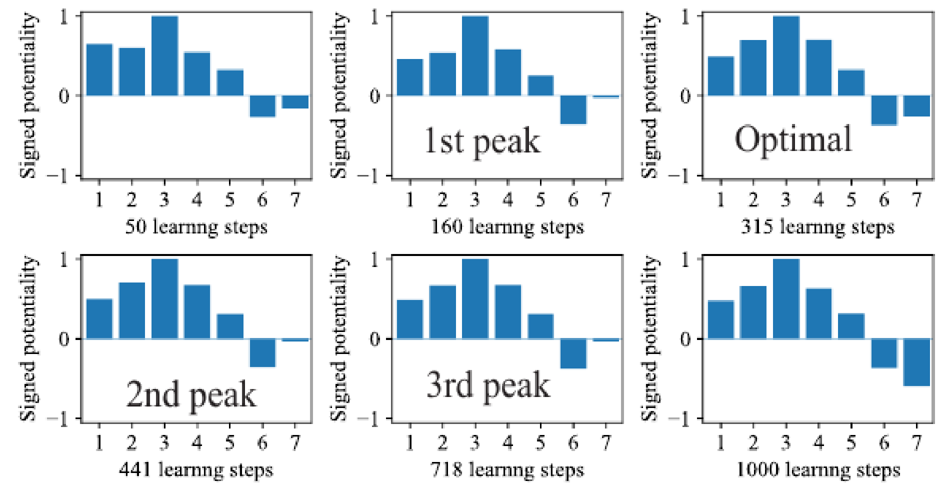
- **Ratio potentiality**
  - Ratio of estimated prototype to the supposed prototype
  - As the ratio increases, similarity to the supposed prototype increases
- **Simple Synchronic potentiality reduction (beta=0)**
  - The wide higher ratio potentialities could be observed in the beginning
- **Neutralization(beta=1.2)**
  - Three peaks with higher ratio potentiality were detected
  - Strong forces to detect the prototype were detected
  - Repeated neutralization had an effect to structuralize networks to have better generalization



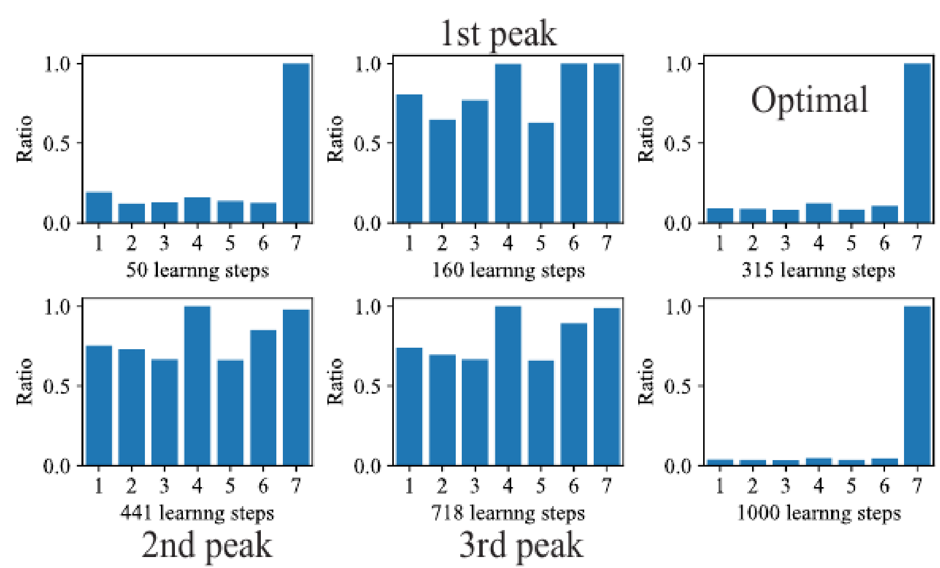
Weights and Ratio Potentialities



Supposed prototype



(a) Signed individual potentiality



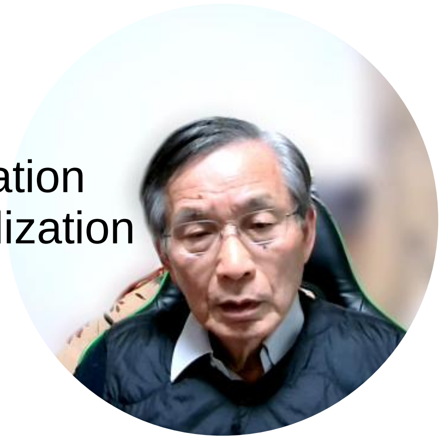
(b) Individual ratio potentiality

- **Non-linear relation**
  - The individual ratio potentiality was higher for only the final non-linear input
  - Only one non-linear input could improve generalization in the optimal case
- **Prototype learning**
  - The prototype detection made it possible to improve generalization by networks close to the prototype

Generalization Accuracy

Peak	Step	Ratio Potent	Divergence	Testing
1st	159	0.836	0.014	0.783
2nd	440	0.805	0.011	0.953
3rd	717	0.806	0.012	0.939
Optimal	314	0.224	0.108	0.959

- **Higher ratio potentiality**
  - Higher ratio potentiality could be observed three times
  - Neutralized potentiality reduction tried to produce networks close to the supposed prototype
- **Better generalization**
  - Detecting the prototypes means better generalization
  - By simpler network configurations, better generalization performance is possible



Summary of Experimental Results

- **Weakening synchronic potentiality reduction**
  - Neutralization had an effect to weaken synchronic potentiality reduction
- **Strong force to detect the prototype**
  - By neutralization, networks were as close as possible to the supposed prototype for the entire learning
- **Simple representation with better generalization**
  - Neutralization produced simpler and linear connection weights, keeping improved generalization

Related and Future Work

- **Simplification**
  - It is necessary to show the existence of simplification forces by many examples
  - For example, we should examine how the simplification force is related to conventional studies

- **Implicit regularization**
  - Neural networks and machine learning

- **U-shaped learning**
  - Cognitive development

- **Least effort principle**
  - Quantitative linguistics