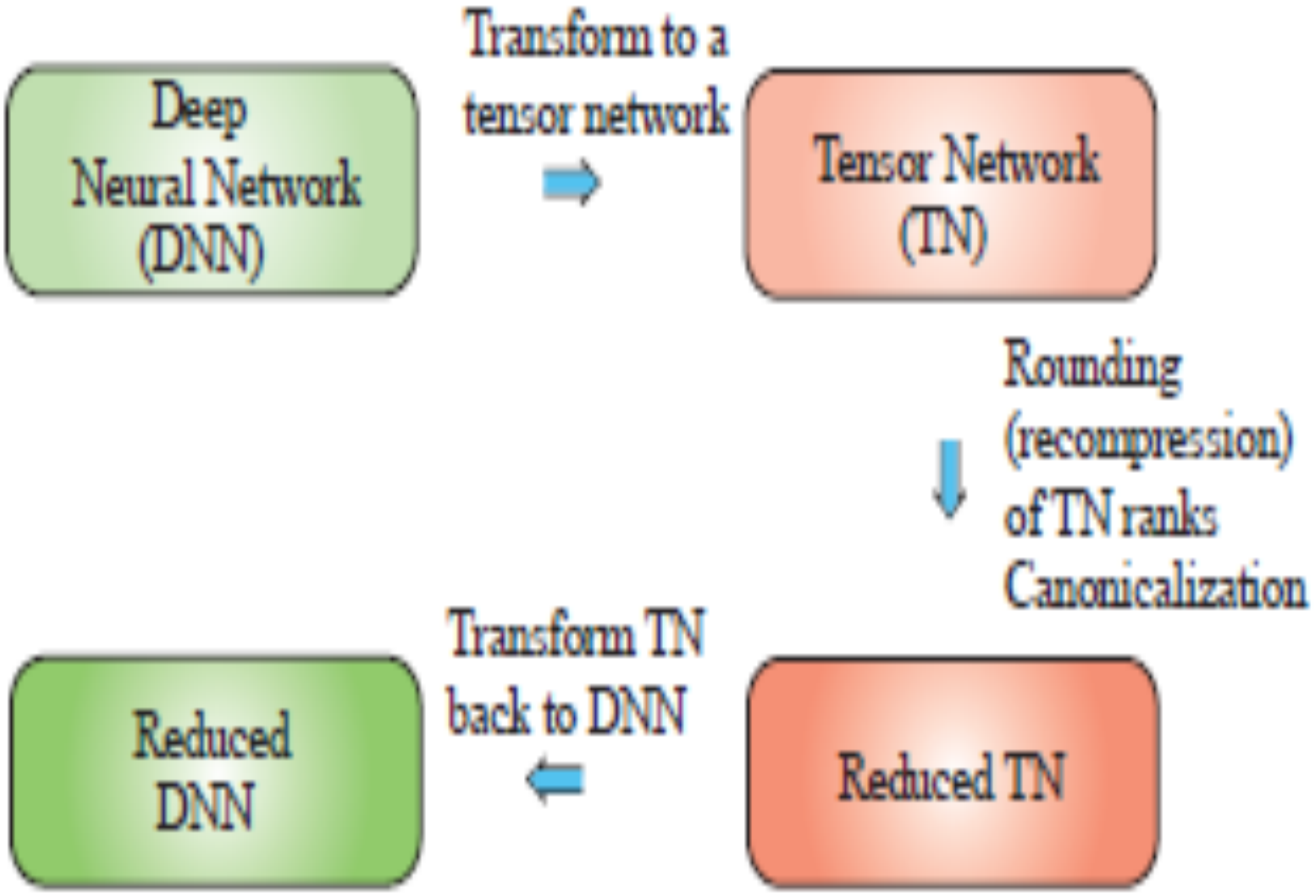
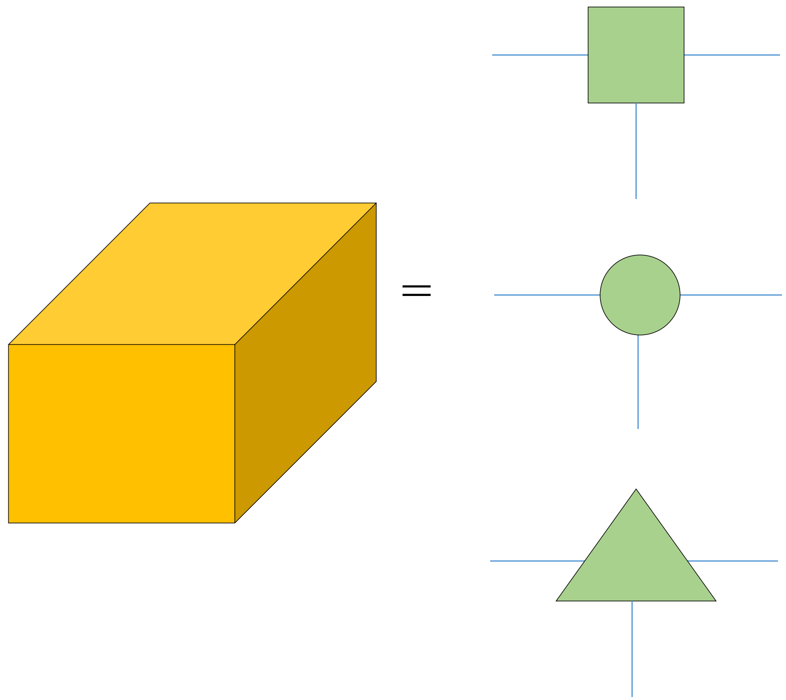
TENSOR NETWORKS
and
DEEP NEURAL NETWORKS
Bridge Between Tensor Networks/ Quantum Physics and Deep Learning
From Theory to Real Applications

Andrzej Cichocki (given by I. Oseledets)
SKOLTECH and RIKEN

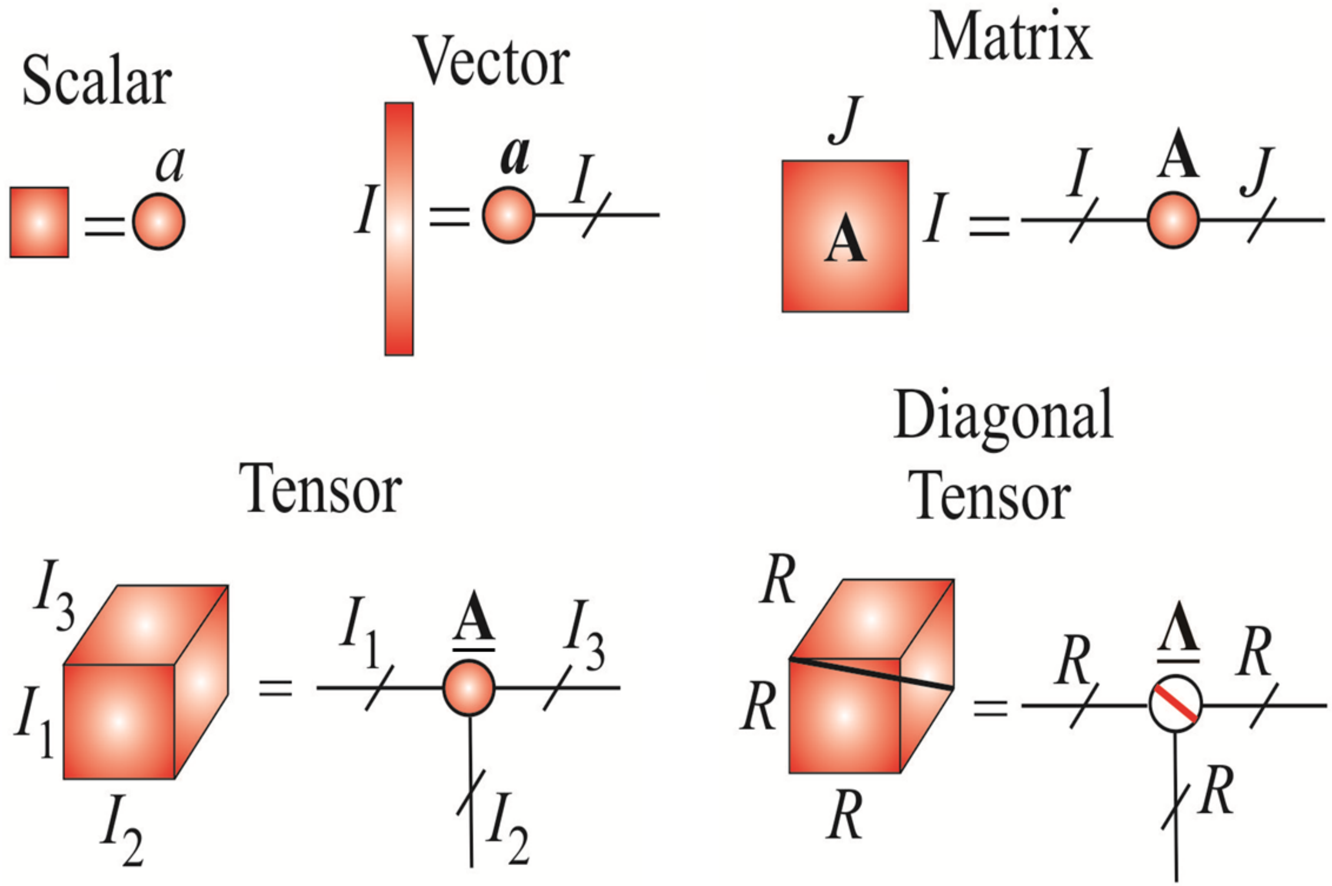
Optimization of Deep Neural Networks (DNNs) Using Tensor Networks



Symbols of 3rd-roder tensors

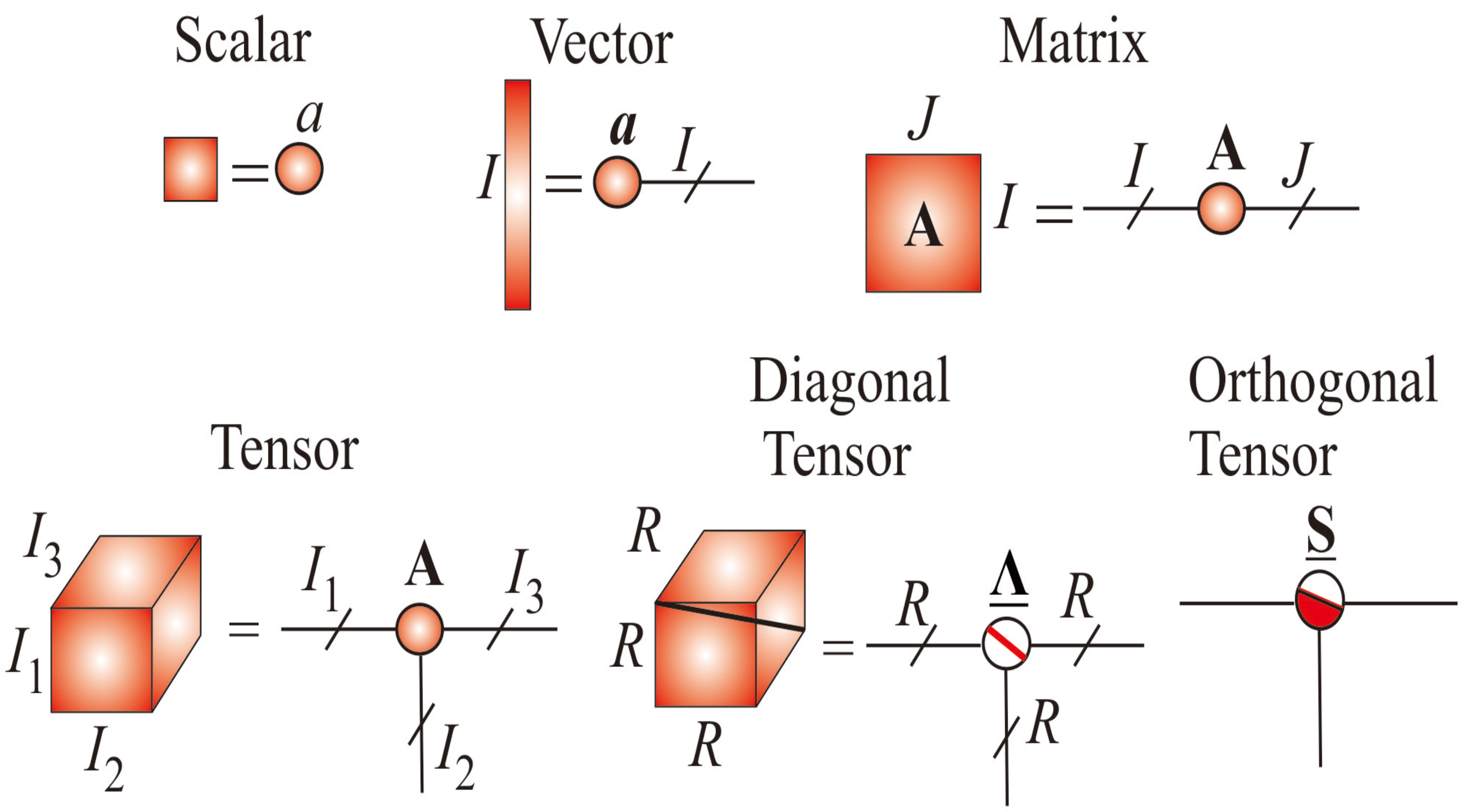


Basic Elements of Tensor Networks

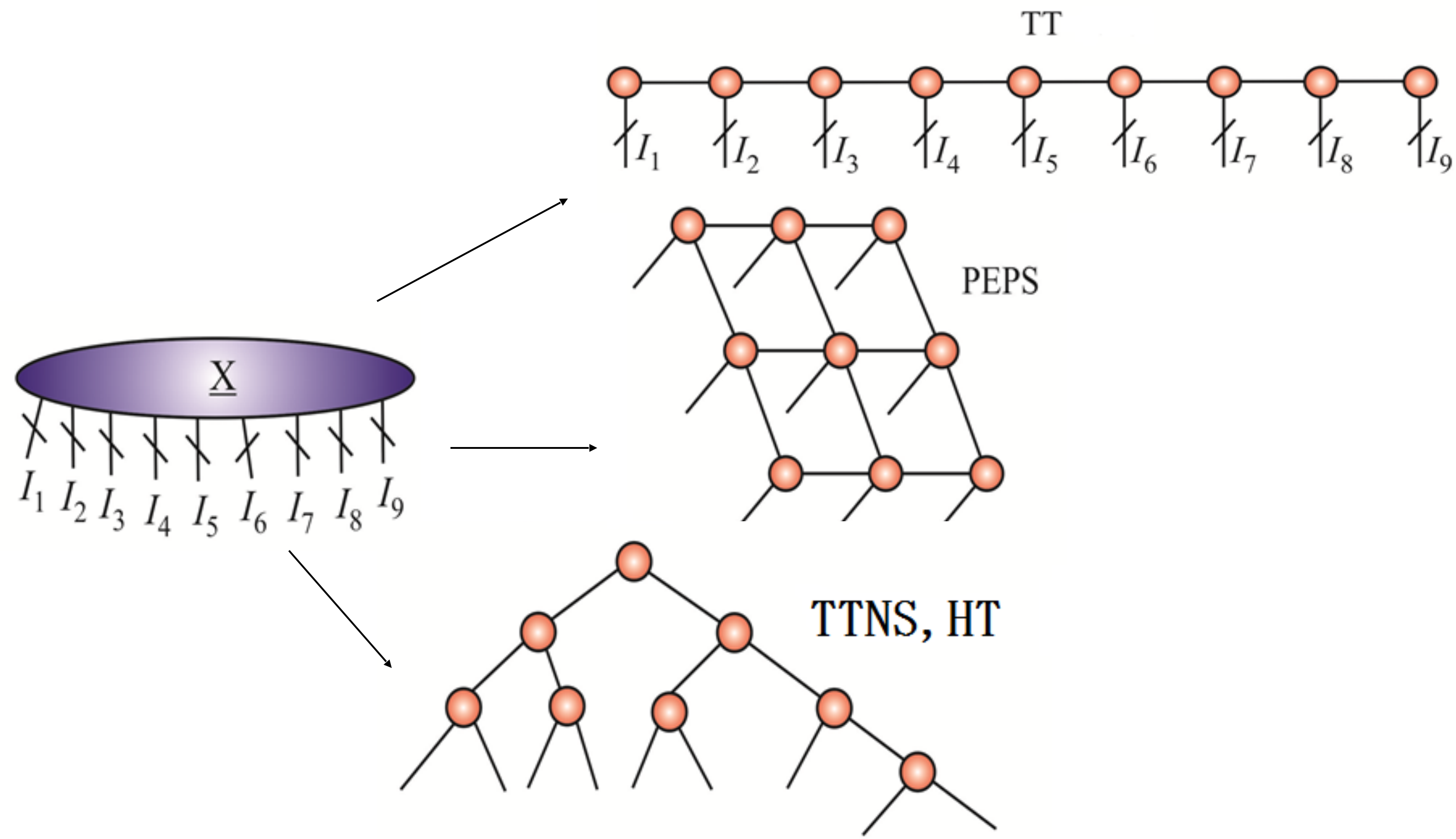


Tensor network diagram can be used for illustrating high-order tensors and multi-linear operations.

Basic Elements of Tensor Networks



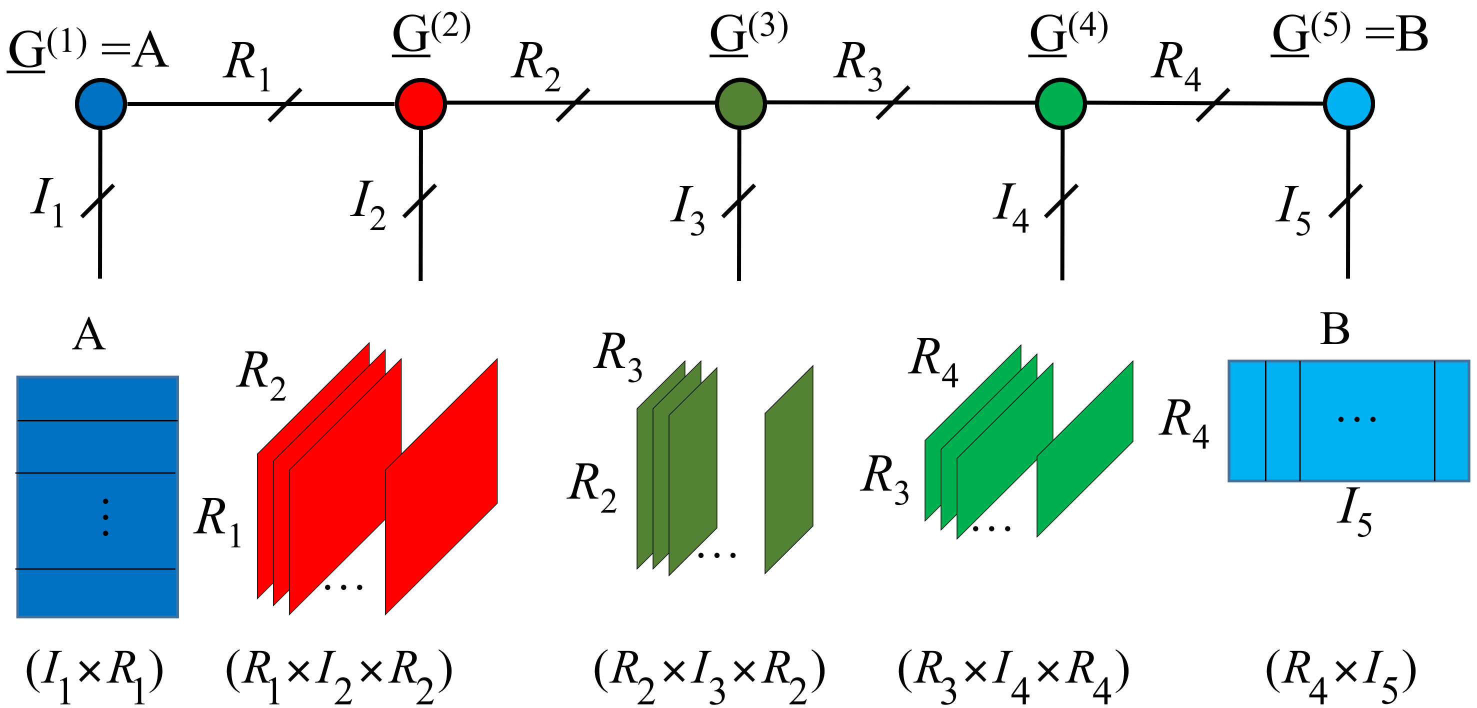
Generalization of tensor decompositions TENSOR NETWORKS
Alleviate the curse of dimensionality problem



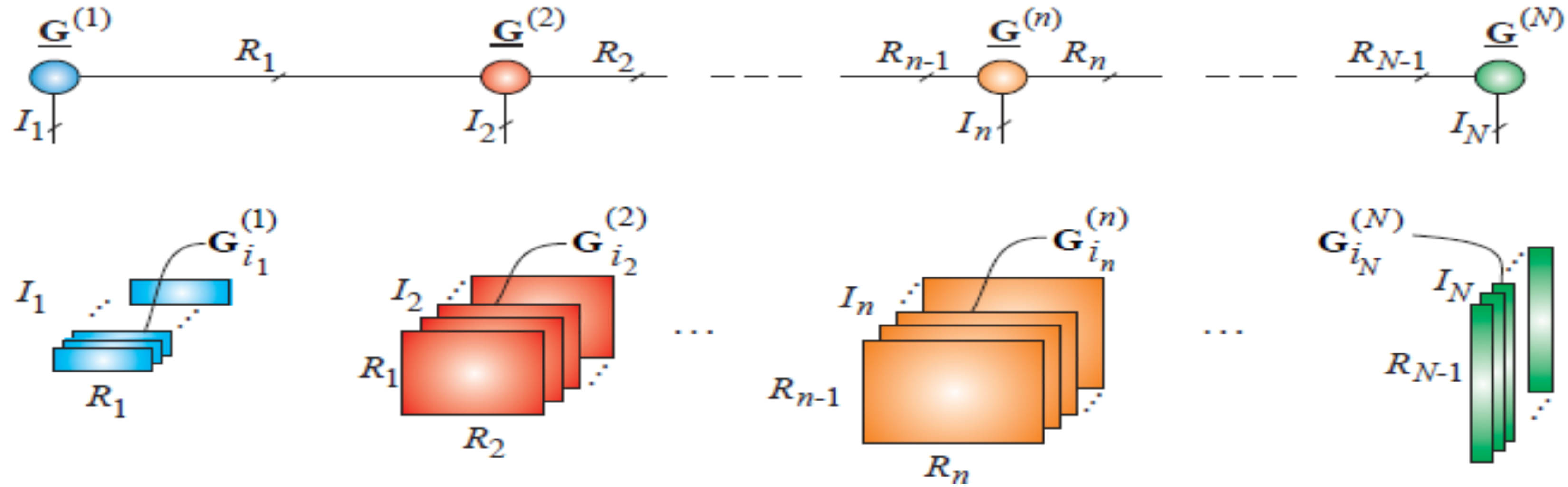
Tensor network models can be applied to cope with the *curse-of-dimensionality* problem in very large-scale TNS and constrained optimizations

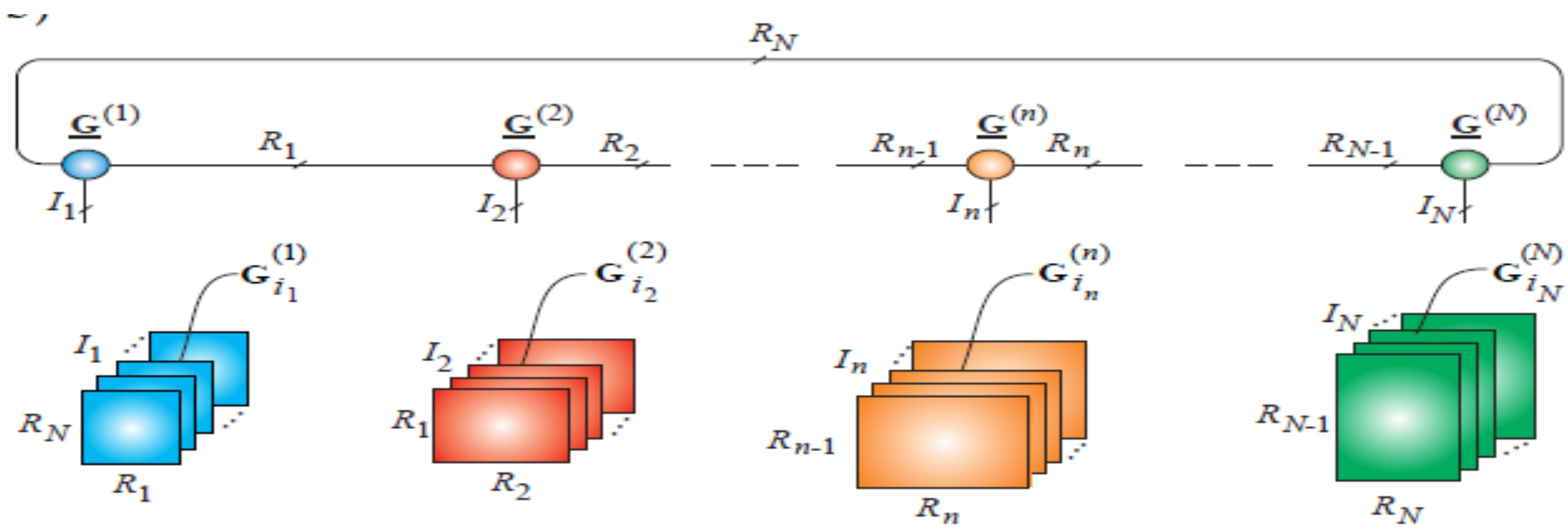
Tensor Train (TT) or Matrix Product State (MPS) with Open Boundary Conditions (OBC)

$$\underline{\mathbf{X}} \cong \mathbf{A} \times_2^1 \underline{\mathbf{G}}^{(2)} \times_3^1 \underline{\mathbf{G}}^{(3)} \times_3^1 \underline{\mathbf{G}}^{(4)} \times_3^1 \mathbf{B}$$

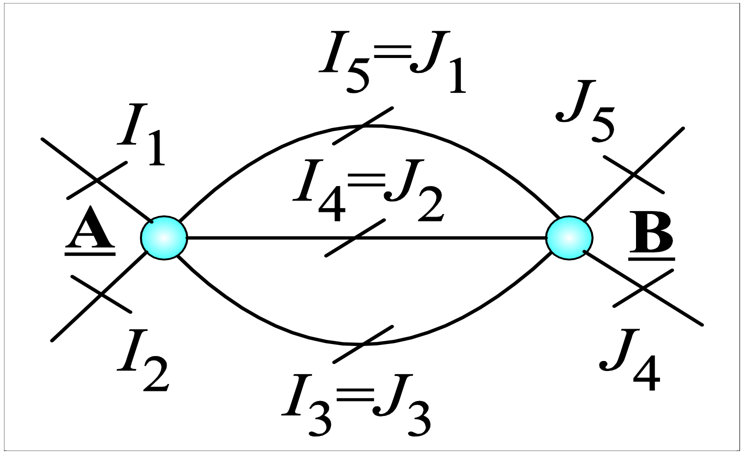


Tensor Train and Tensor Chain (MPS with OBC and PBC)





Tensor/matrix contractions



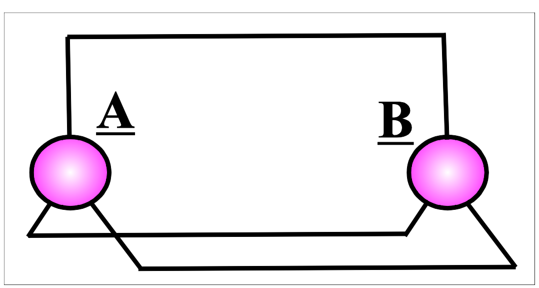
$$\underline{\mathbf{X}} = \underline{\mathbf{A}} \times_{3,4,5}^{3,2,1} \underline{\mathbf{B}}$$

$$x_{i_1 i_2 j_3 j_4} = \sum_{i_3, i_4, i_5}^{I_3, I_4, I_5} a_{i_1 i_2 i_3 i_4 i_5} b_{i_5 i_4 i_3 j_4 j_5}$$

$$x_{i_1 i_2 j_3 j_4} = \sum_{i_3, i_4, i_5}^{I_3, I_4, I_5} \phi(a_{i_1 i_2 i_3 i_4 i_5}, b_{i_5 i_4 i_3 j_4 j_5}),$$

$$\phi(a, b) = \max(0, ab)$$

Inner product



$$\underline{\mathbf{X}} = \underline{\mathbf{A}} \times_{1,2,3}^{1,2,3} \underline{\mathbf{B}} = \underline{\mathbf{A}} \times \underline{\mathbf{B}}$$

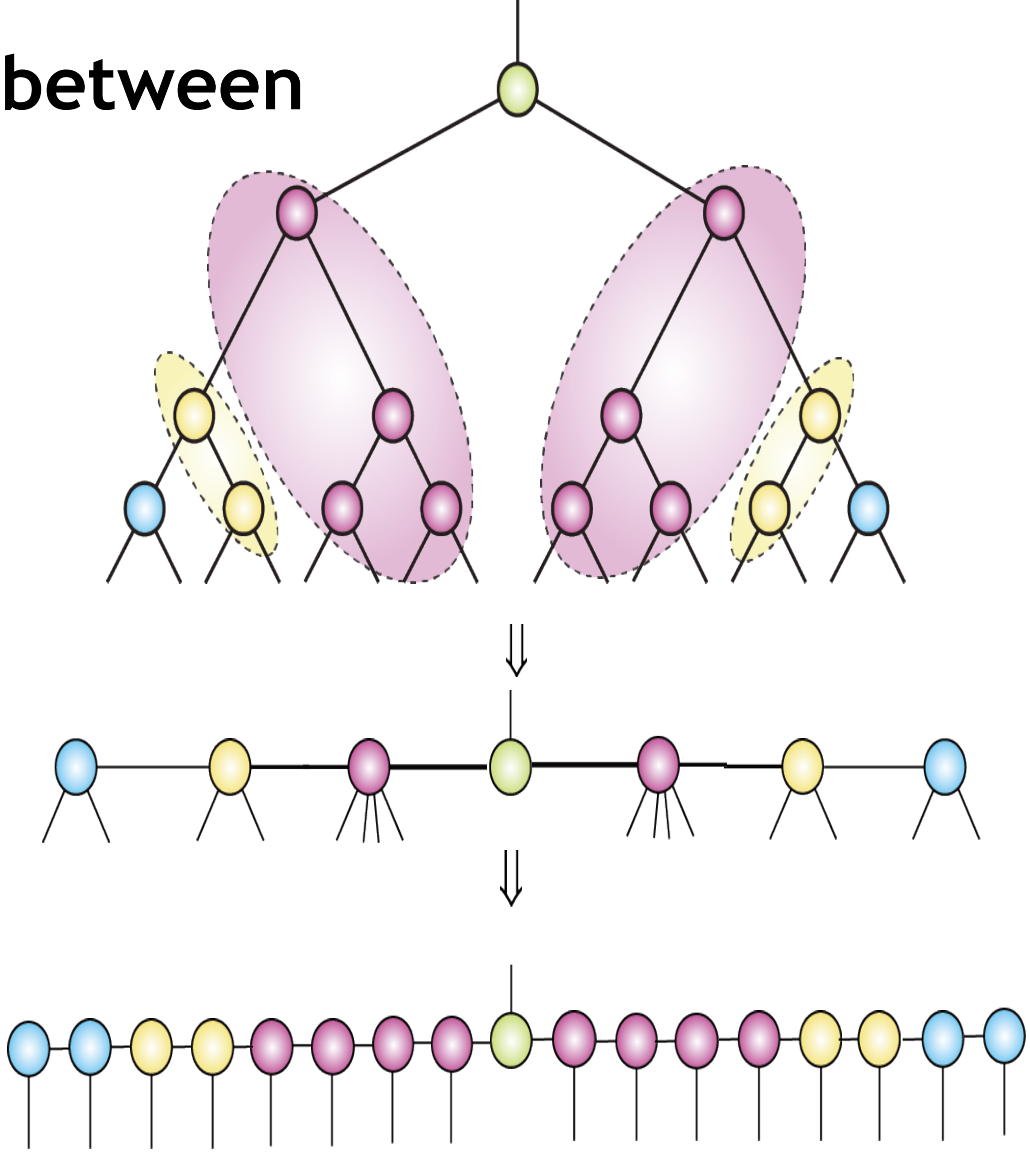
$$x = \sum_{i_1, i_2, i_2}^{I_1, I_2, I_3} a_{i_1 i_2 i_3} \; b_{i_1 i_2 i_3}$$

Equivalence between TNs

HT (Hierarchical Tucker)
Cohen -Shashua

Tensor Train with core tensors with variable orders

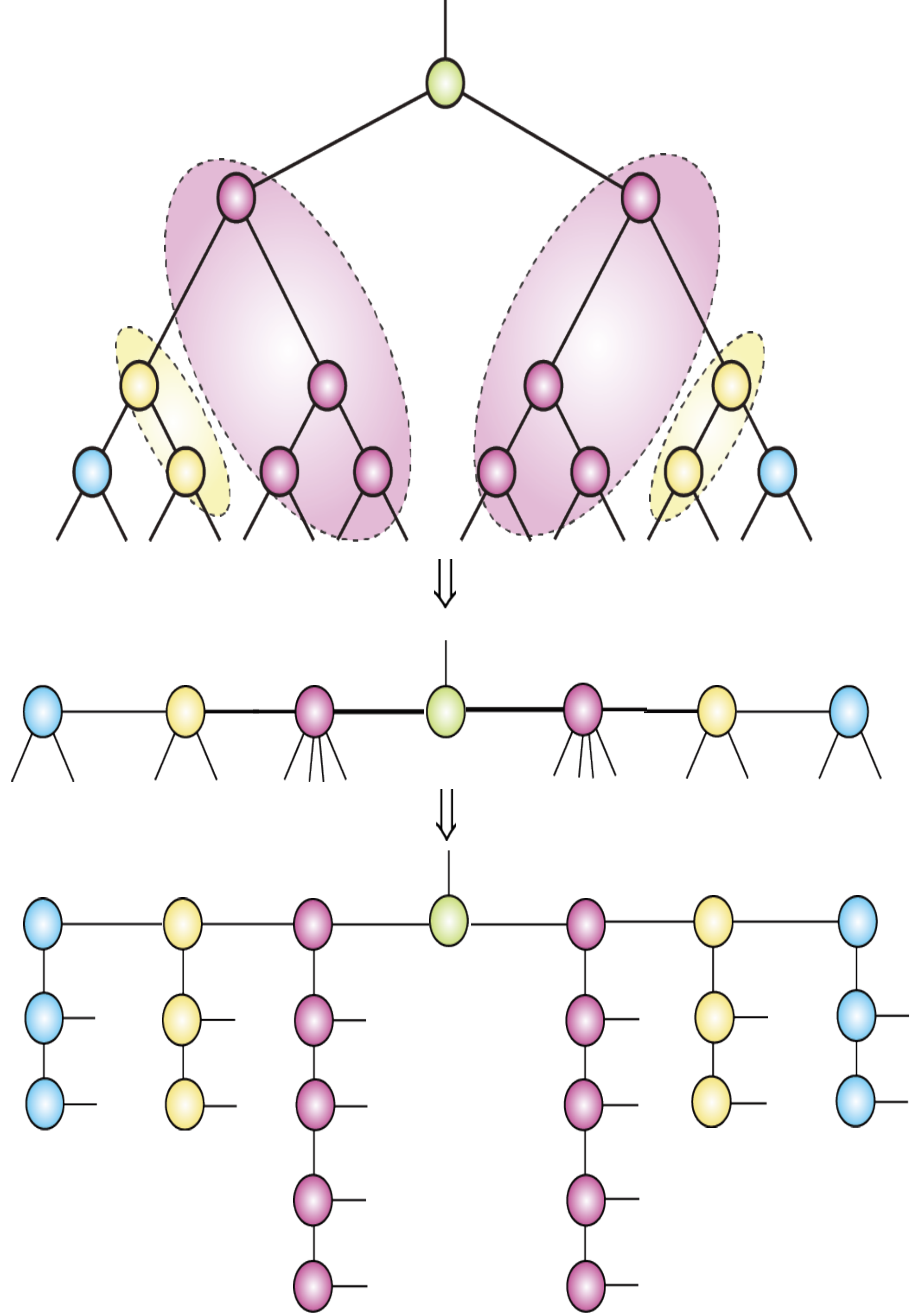
Standard Tensor Train TT/QTT



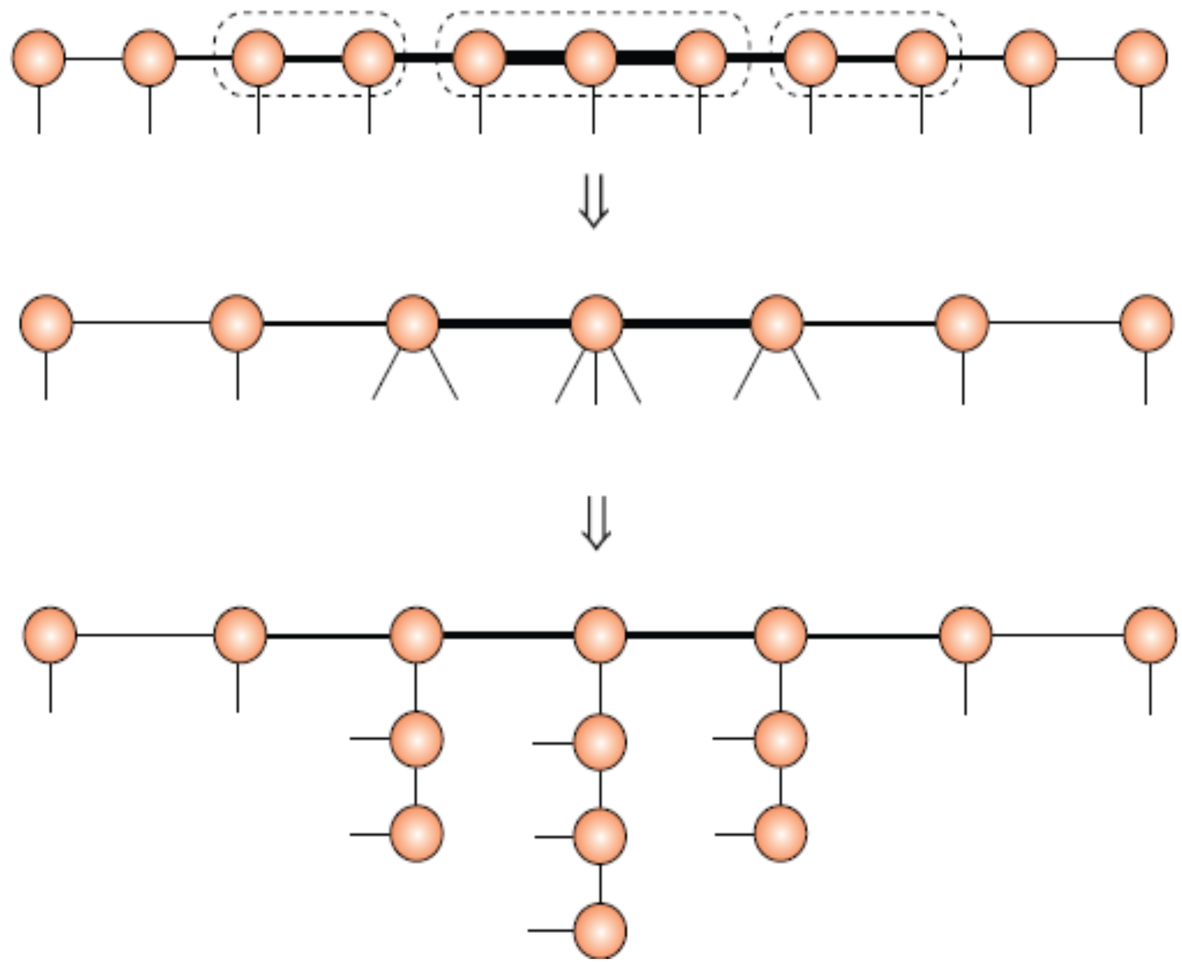
HT (Hierarchical Tucker)

Tensor Train with core tensors with variable orders

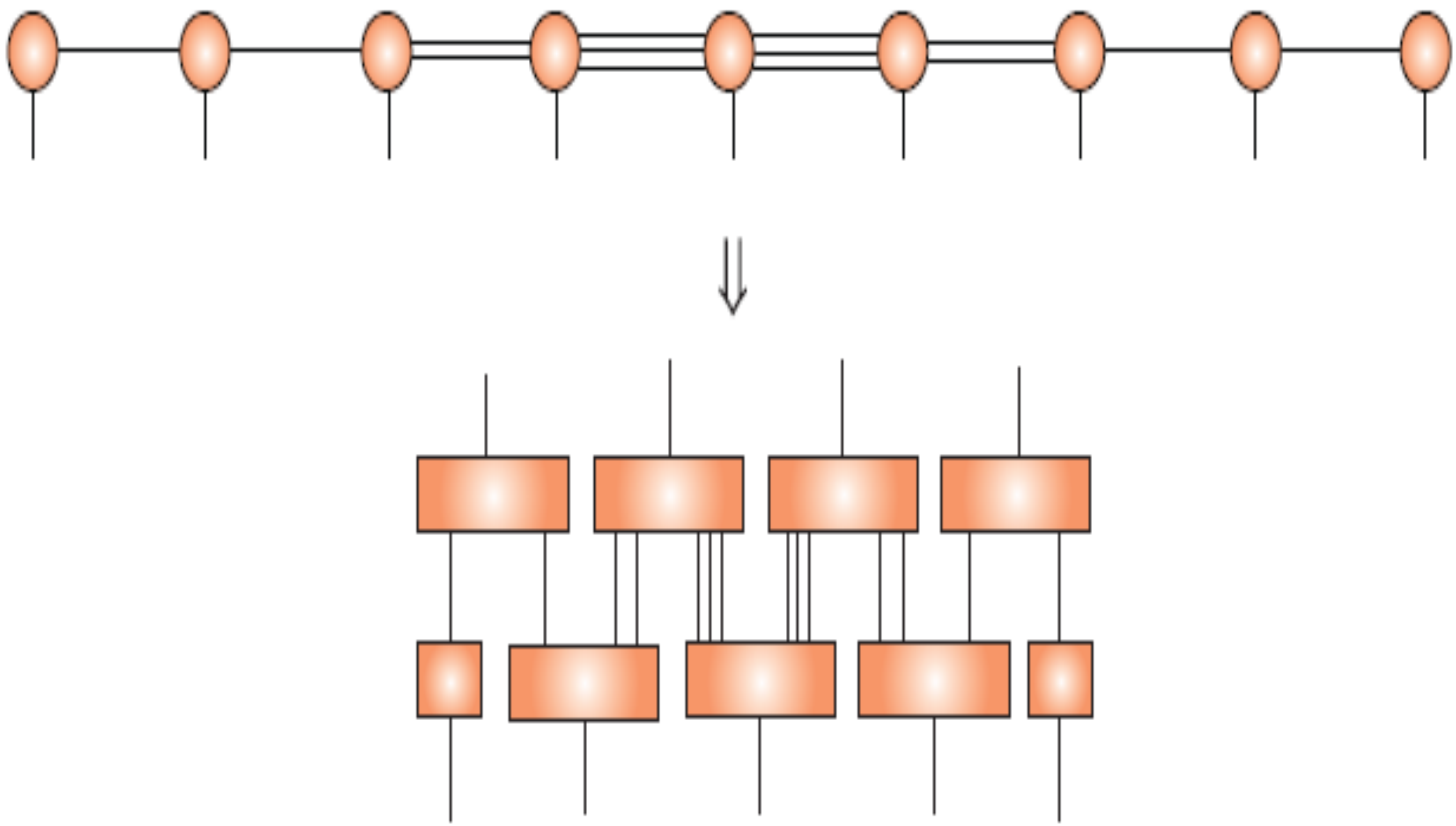
Fork Tensor Train or Fork MPS



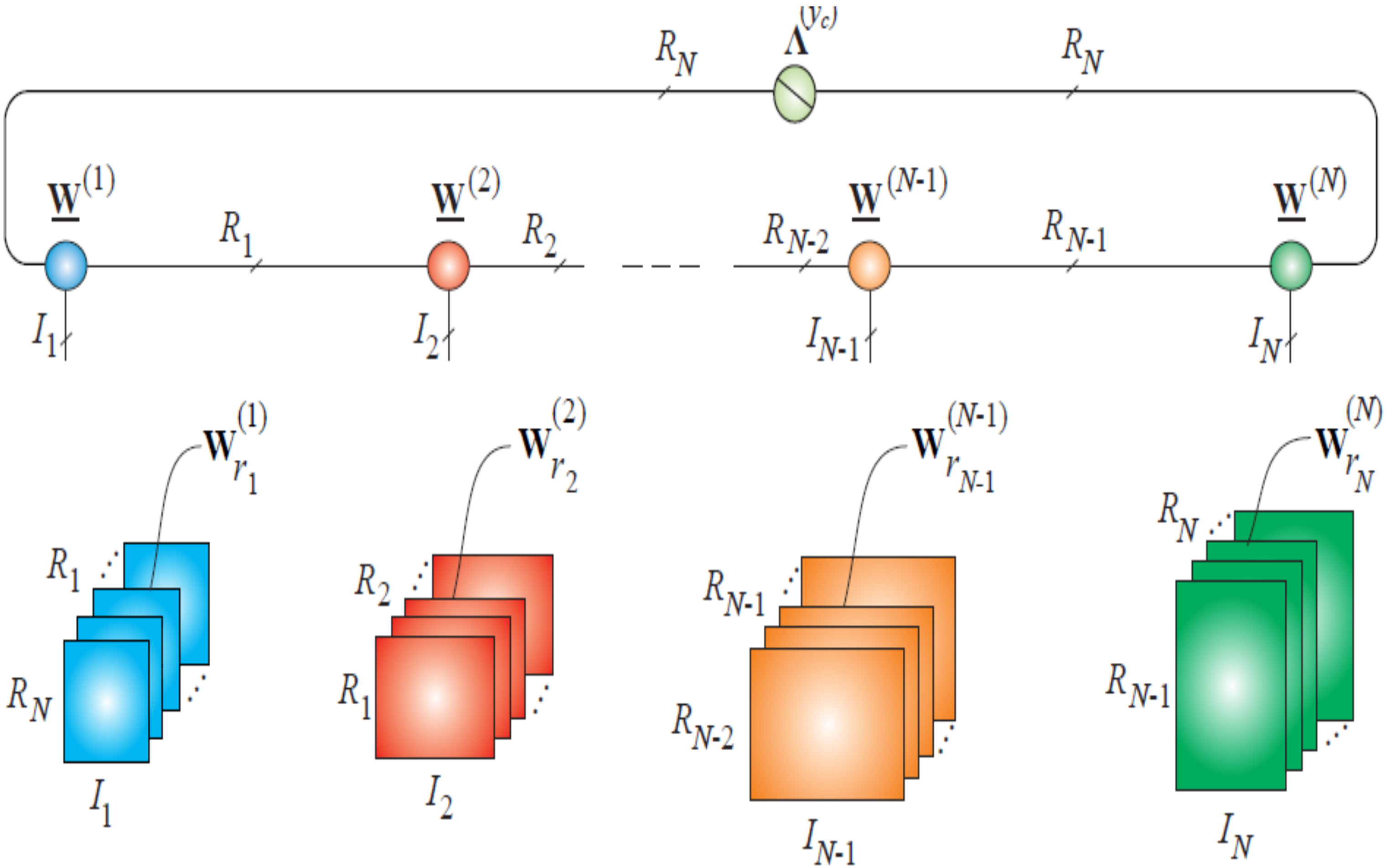
Network in Network (TT in TT)



Network in Network (TT in TT)



Tensor Train (MPS) Graphical representation via matrices (slices)



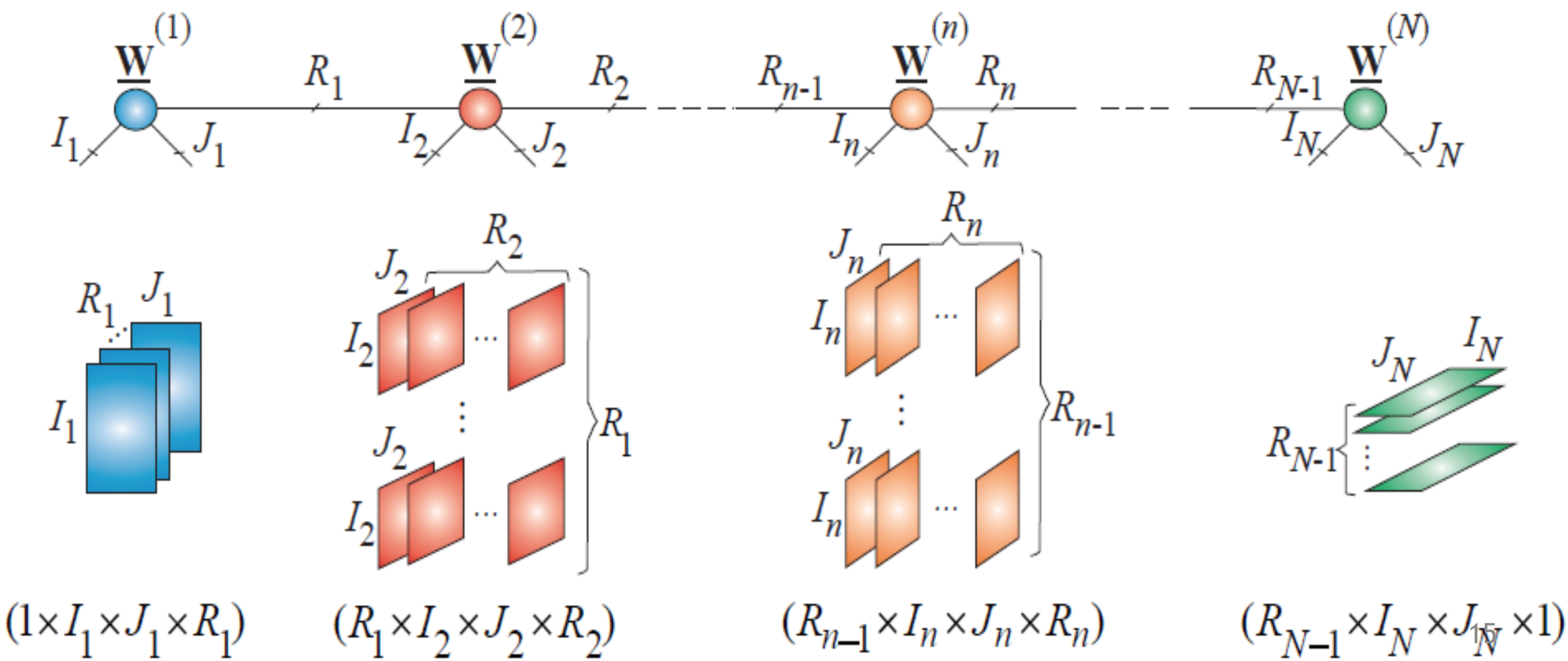
$$\underline{\mathbf{W}}_{y_c} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} \lambda^{(y_c)}_{r_N} \left(\mathbf{w}^{(1)}_{r_N,r_1} \circ \mathbf{w}^{(2)}_{r_1,r_2} \circ \cdots \circ \mathbf{w}^{(N)}_{r_{N-1},r_N}\right)$$

A. Cichocki

A. Cichocki

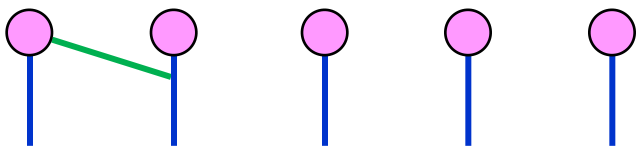
$$h_{y_c} = \sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}\sum_{j_1=1}^{J_1}\cdots\sum_{j_N=1}^{J_N} \underline{\mathbf{W}}_{y_c}(i_1,\ldots,i_N,j_1,\ldots,j_N)\prod_{n=1}^{N} f_{\theta_{i_n,j_n}}(\mathbf{x}_n)$$

$$\underline{\mathbf{W}}_{y_c} = \sum_{r_1=1}^{R_1}\cdots\sum_{r_{N-1}=1}^{R_{N-1}} \lambda^{(y_c)}_{r_{N-1}} (\mathbf{W}^{(1)}_{1,r_1}\circ_\rho \mathbf{W}^{(2)}_{r_1,r_2}\circ_\rho\cdots\circ_\rho \mathbf{W}^{(N-1)}_{r_{N-1},1})$$

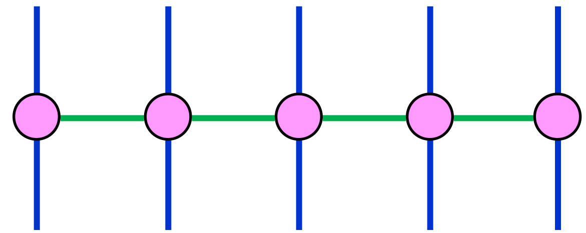


Extended TT decompositions
2D and 3D Tensor Networks PEPS and PEPO

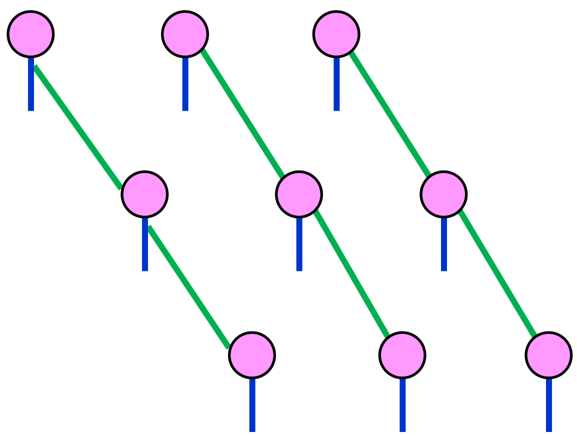
Vector TT -- MPS



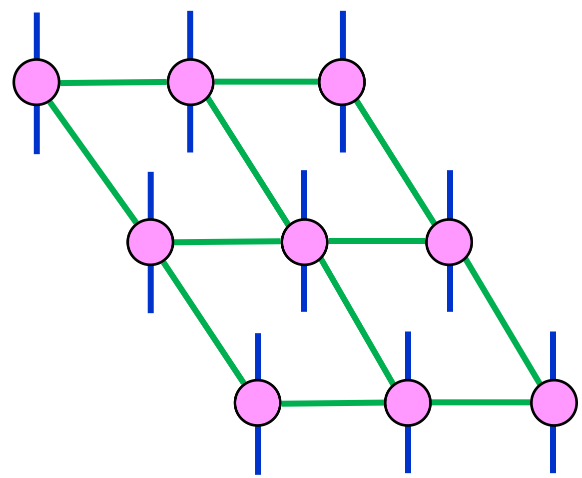
Matrix TT -- MPO



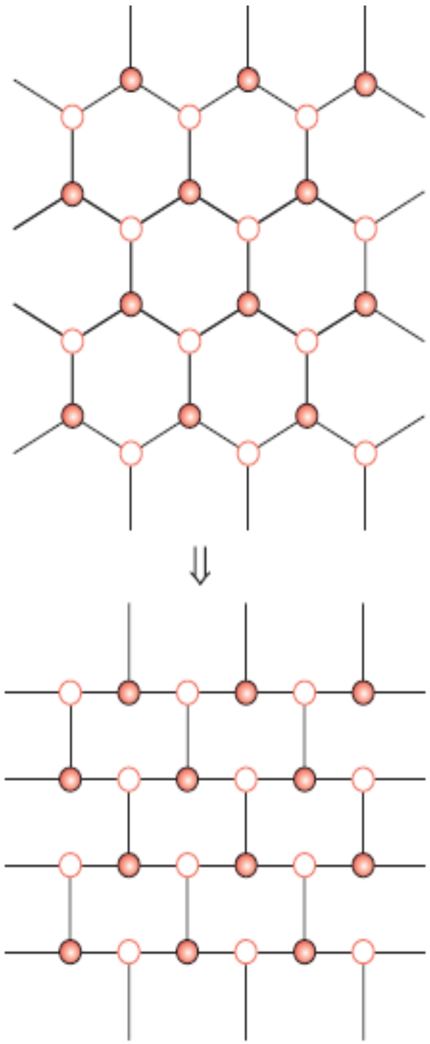
PEPS

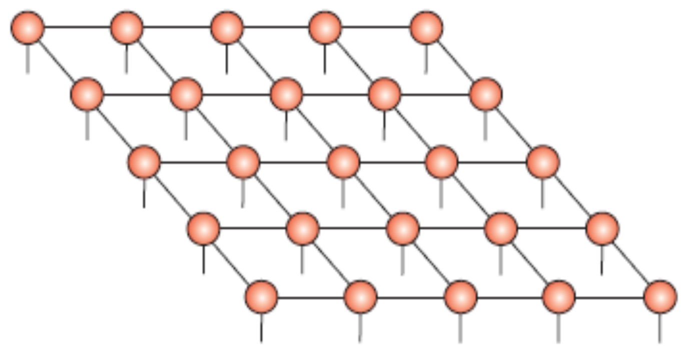


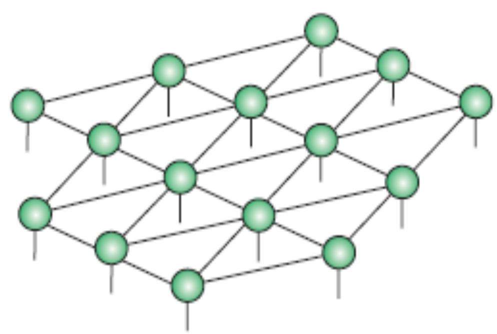
PEPO



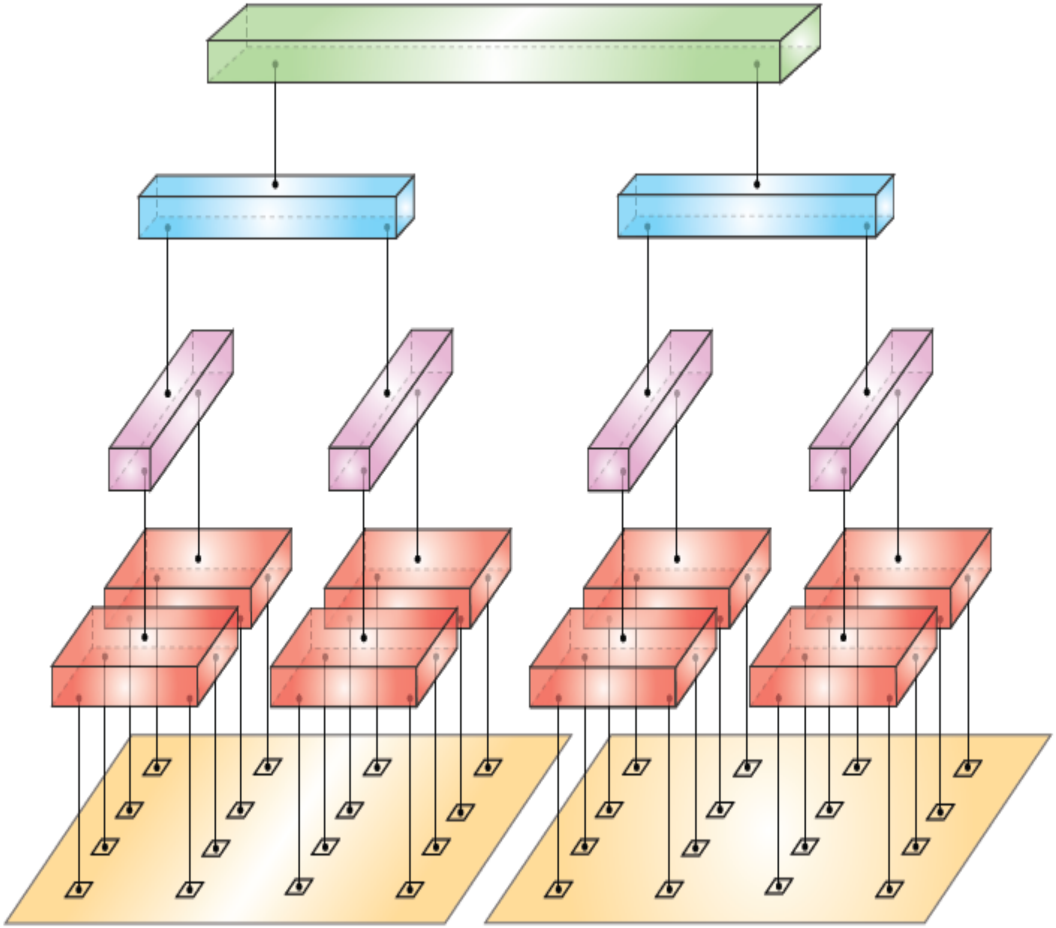
Other TN Architectures
PEPS, PEPO, Honey-Comb Lattice ?



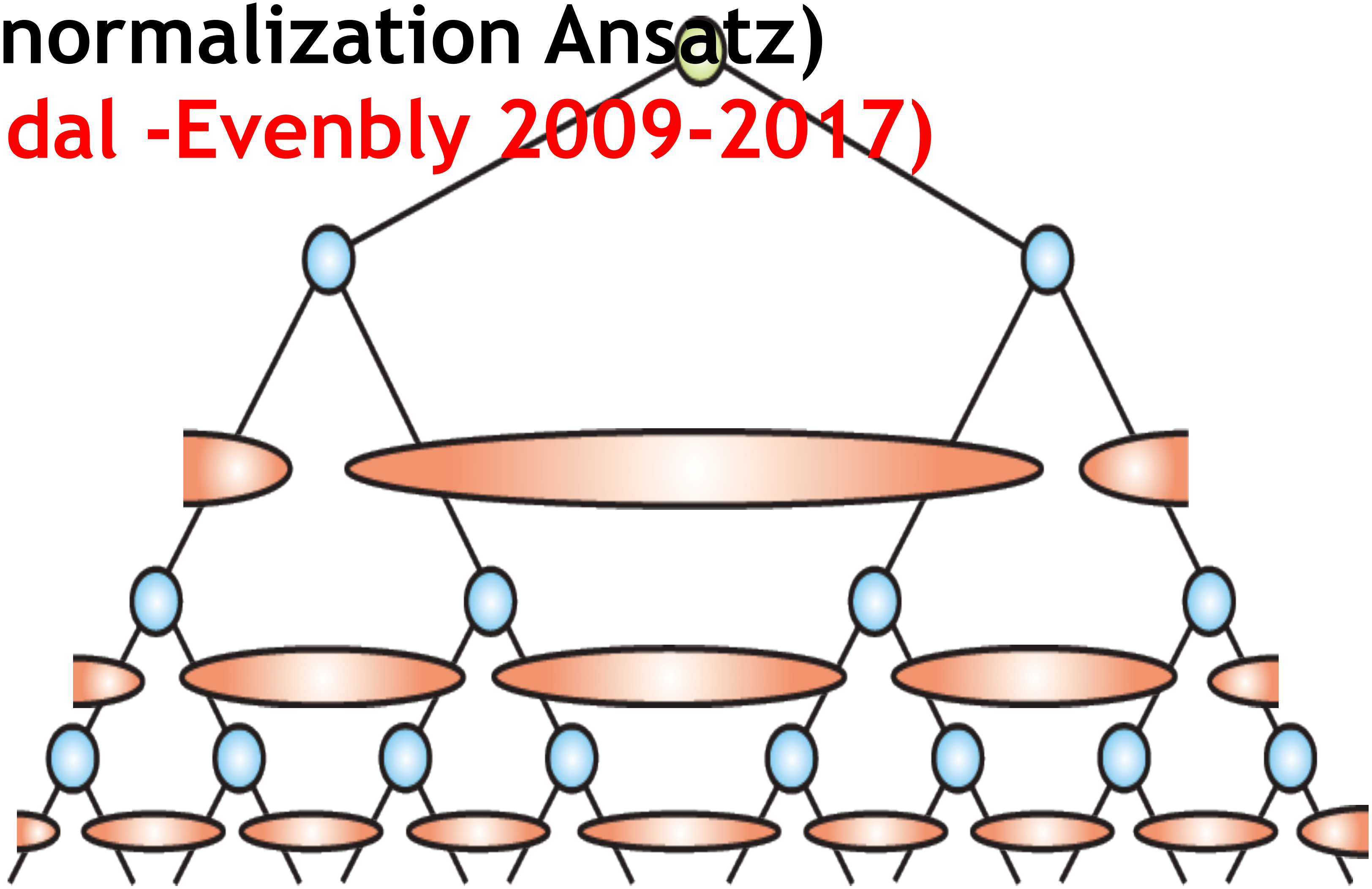




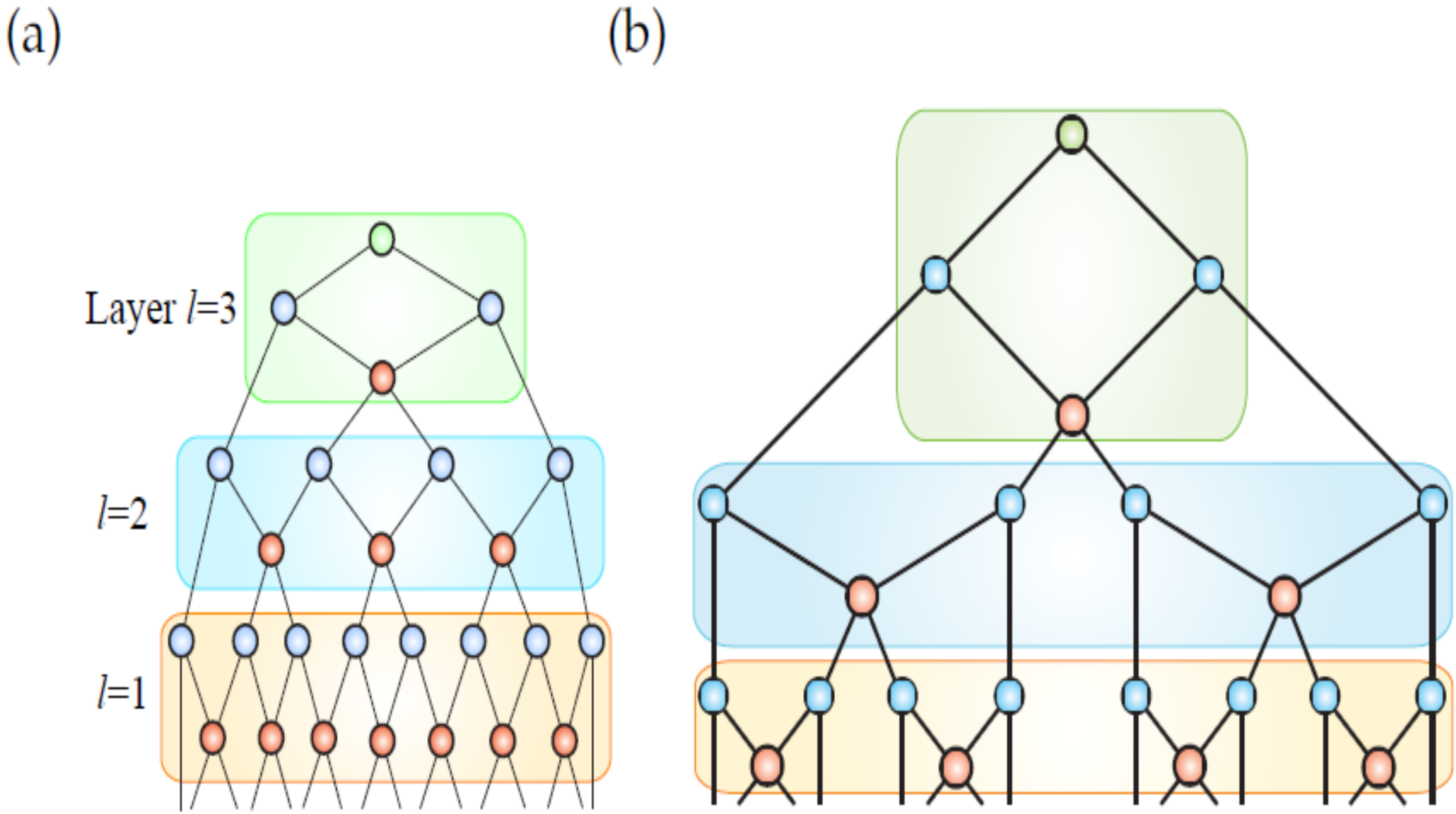
TTNS (Tree Tensor Network States)



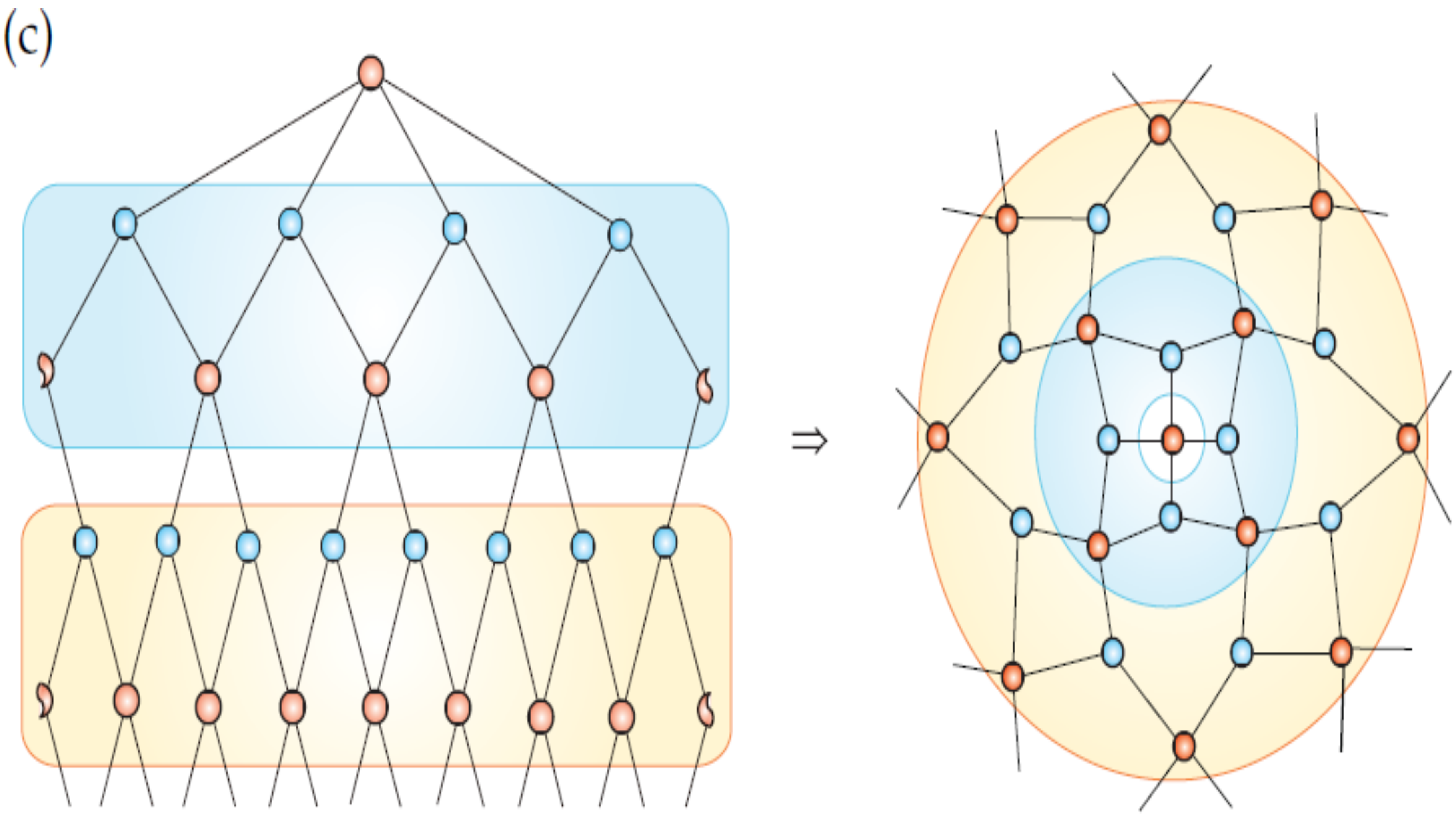
MERA (Multiscale Entanglement Renormalization Ansatz) (Vidal -Evenbly 2009-2017)



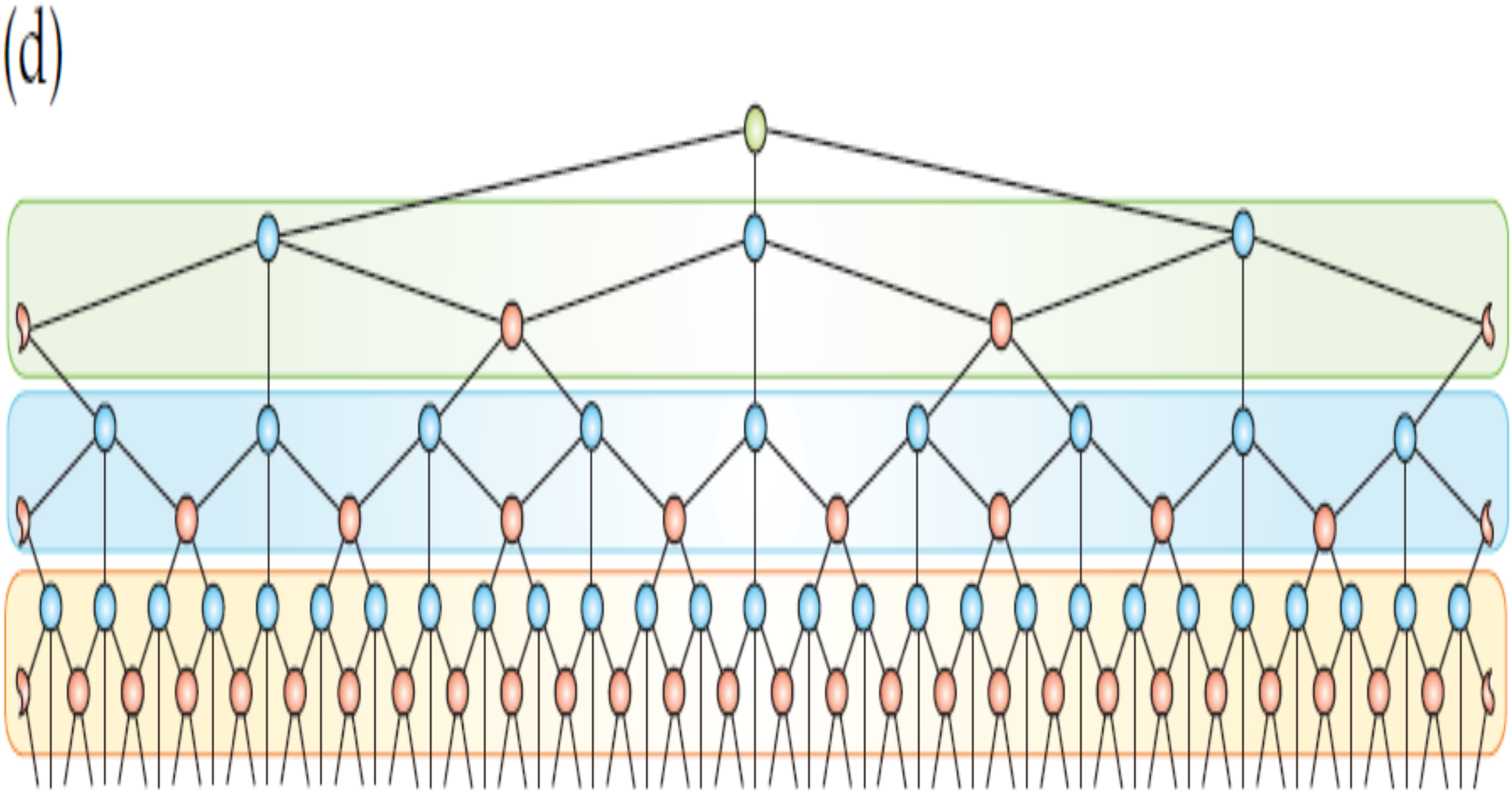
MERA (The Multiscale Entanglement Renormalization Ansatz)



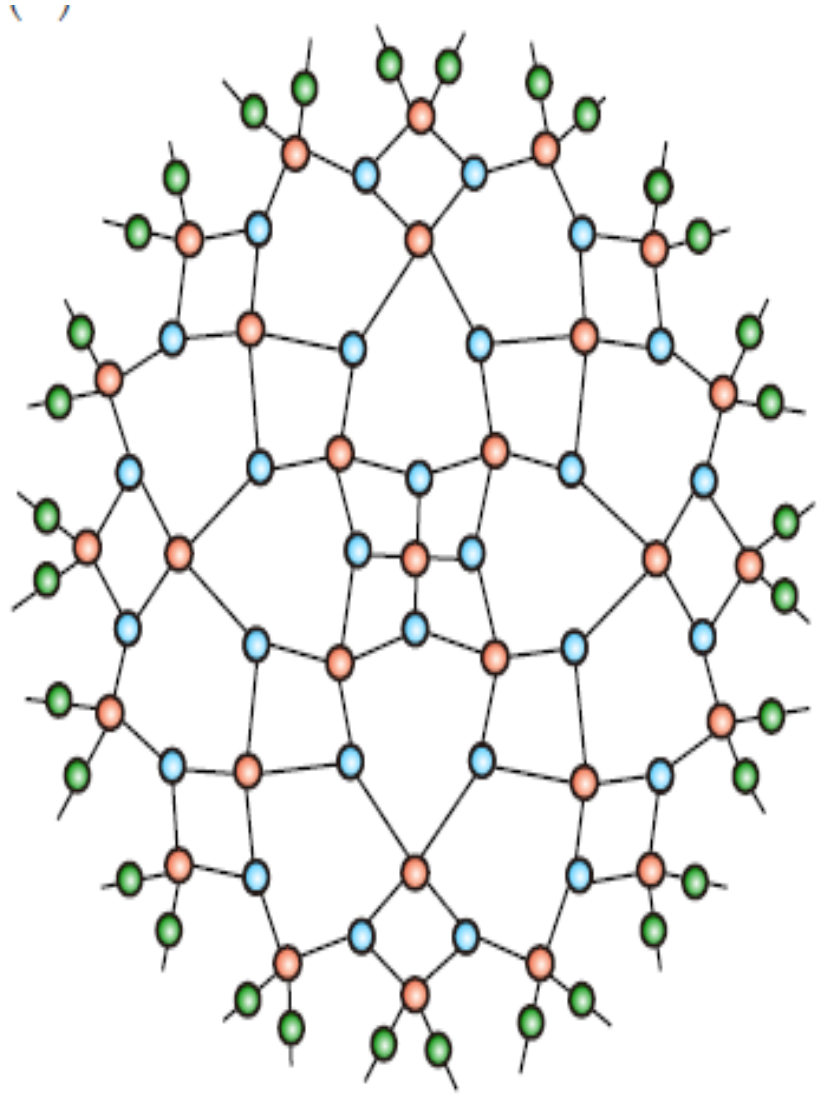
Binary MERA with PBC



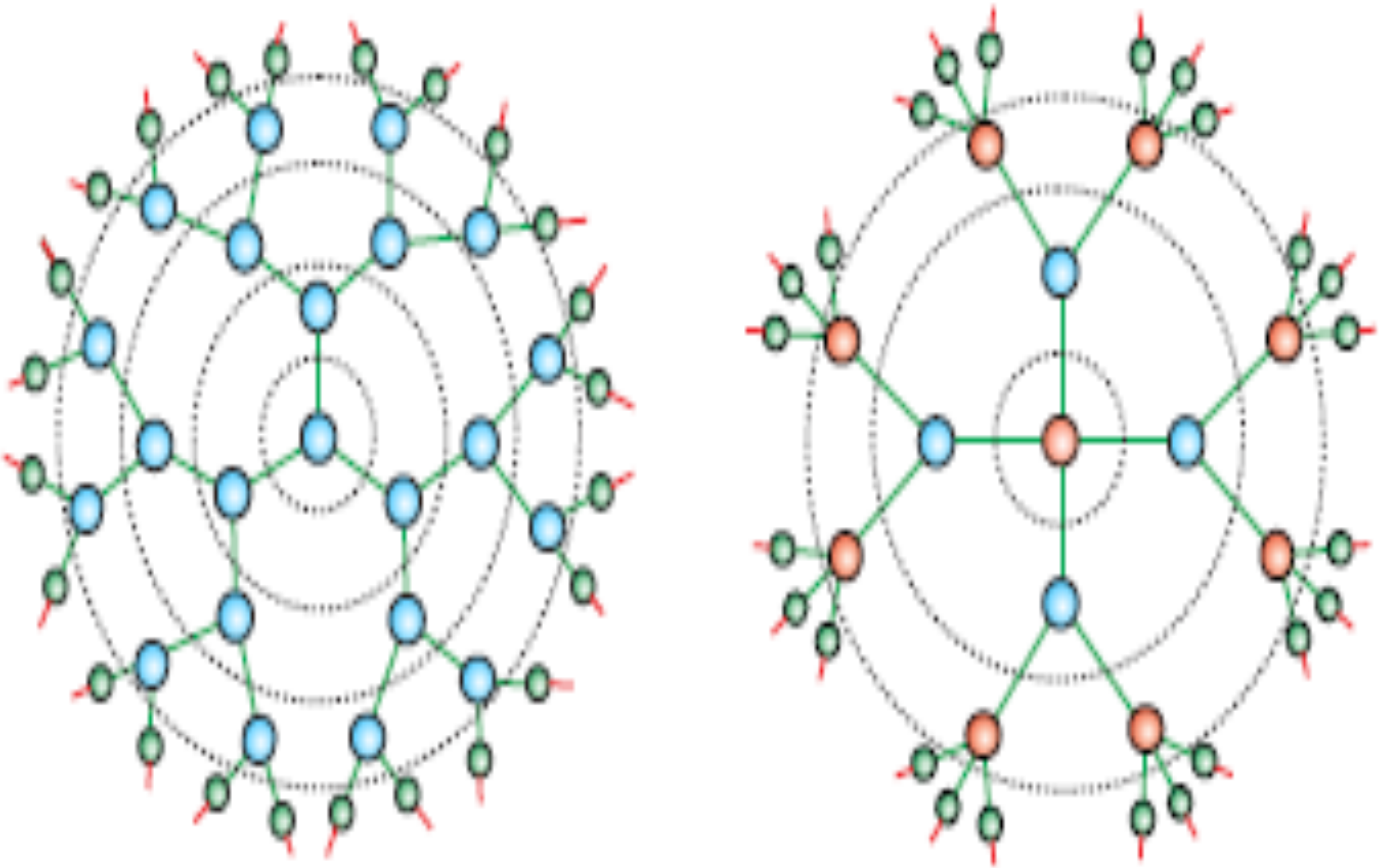
Ternary MERA



MERA for 32-order tensor



TTNs with fixed and variable order of core tensors

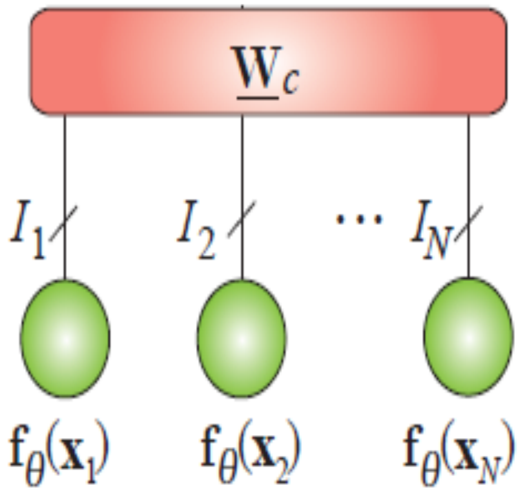


Exponential Machine and TT

$$y = h_c(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \underline{\mathbf{W}}_c(i_1, \ldots, i_N) \prod_{n=1}^{N} f_{\theta_{i_n}}(\mathbf{x}_n),$$

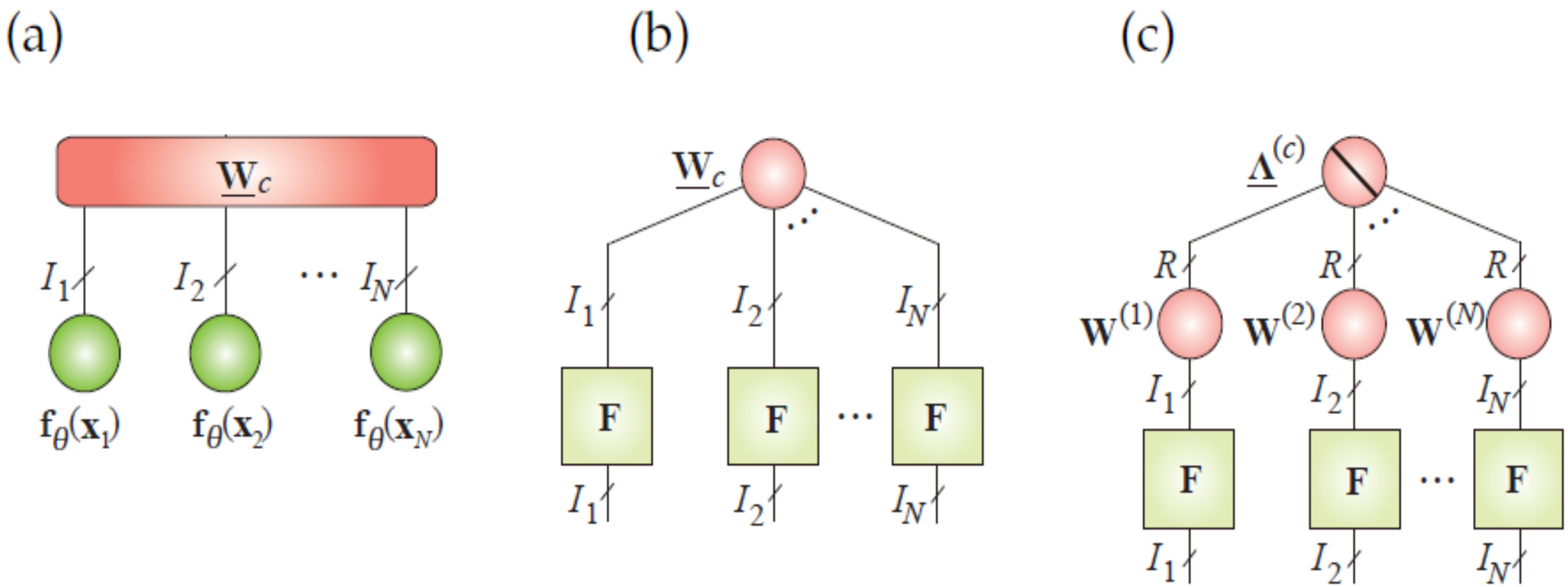
$$h_c(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \langle \underline{\mathbf{W}}_c, \underline{\mathbf{F}} \rangle = \underline{\mathbf{W}}_c \,\bar{\times}_1\, \mathbf{f}_\theta(\mathbf{x}_1) \,\bar{\times}_2\, \mathbf{f}_\theta(\mathbf{x}_2) \cdots \bar{\times}_N\, \mathbf{f}_\theta(\mathbf{x}_N)$$

$$\underline{\mathbf{F}} = \mathbf{f}_\theta(\mathbf{x}_1) \circ \mathbf{f}_\theta(\mathbf{x}_2) \circ \cdots \circ \mathbf{f}_\theta(\mathbf{x}_N) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}.$$



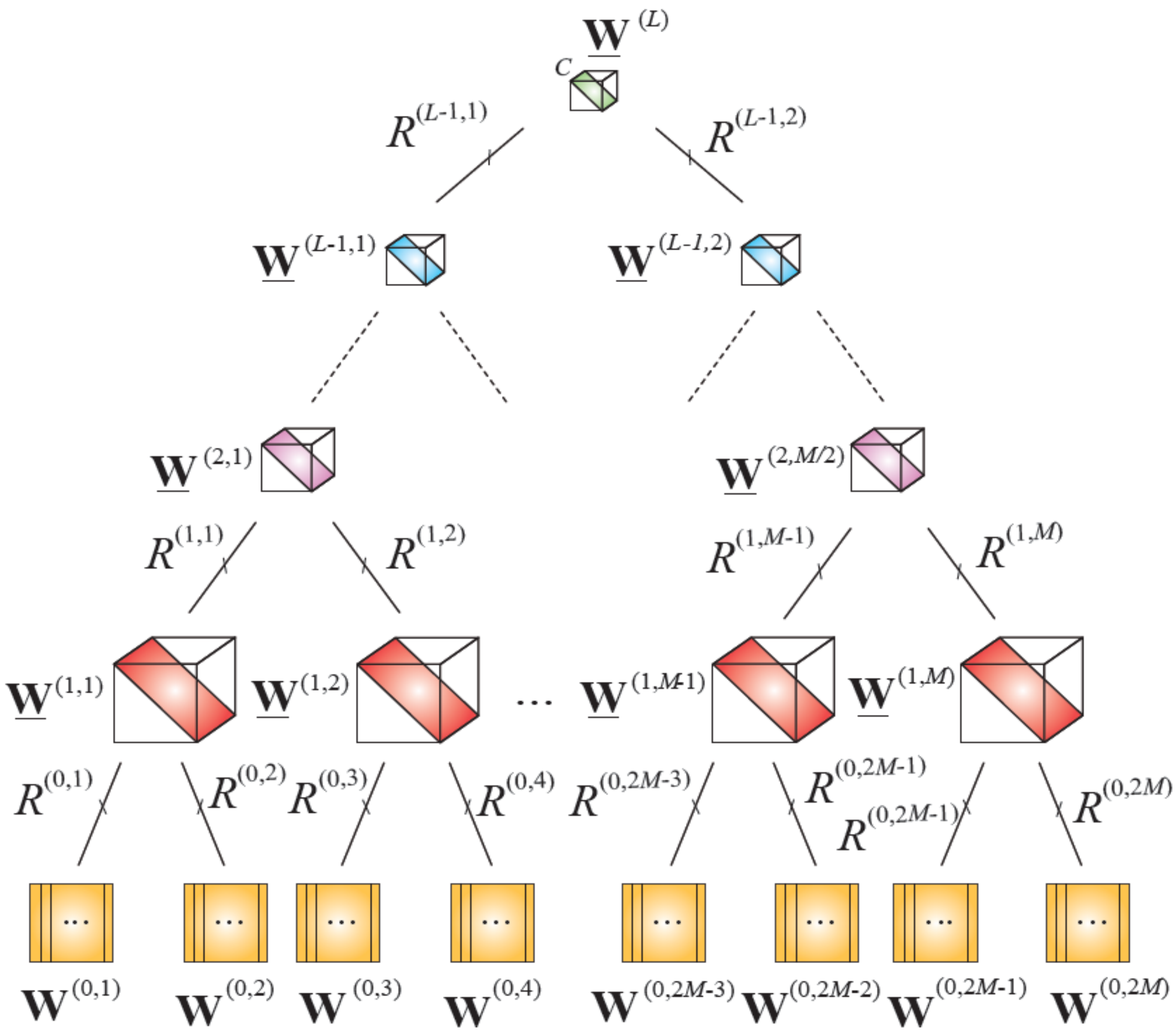
CP, Tucker, HT, TT, Exponential Machine

$$y = h_c(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \underline{\mathbf{W}}_c(i_1, \ldots, i_N) \prod_{n=1}^{N} f_{\theta_{i_n}}(\mathbf{x}_n),$$

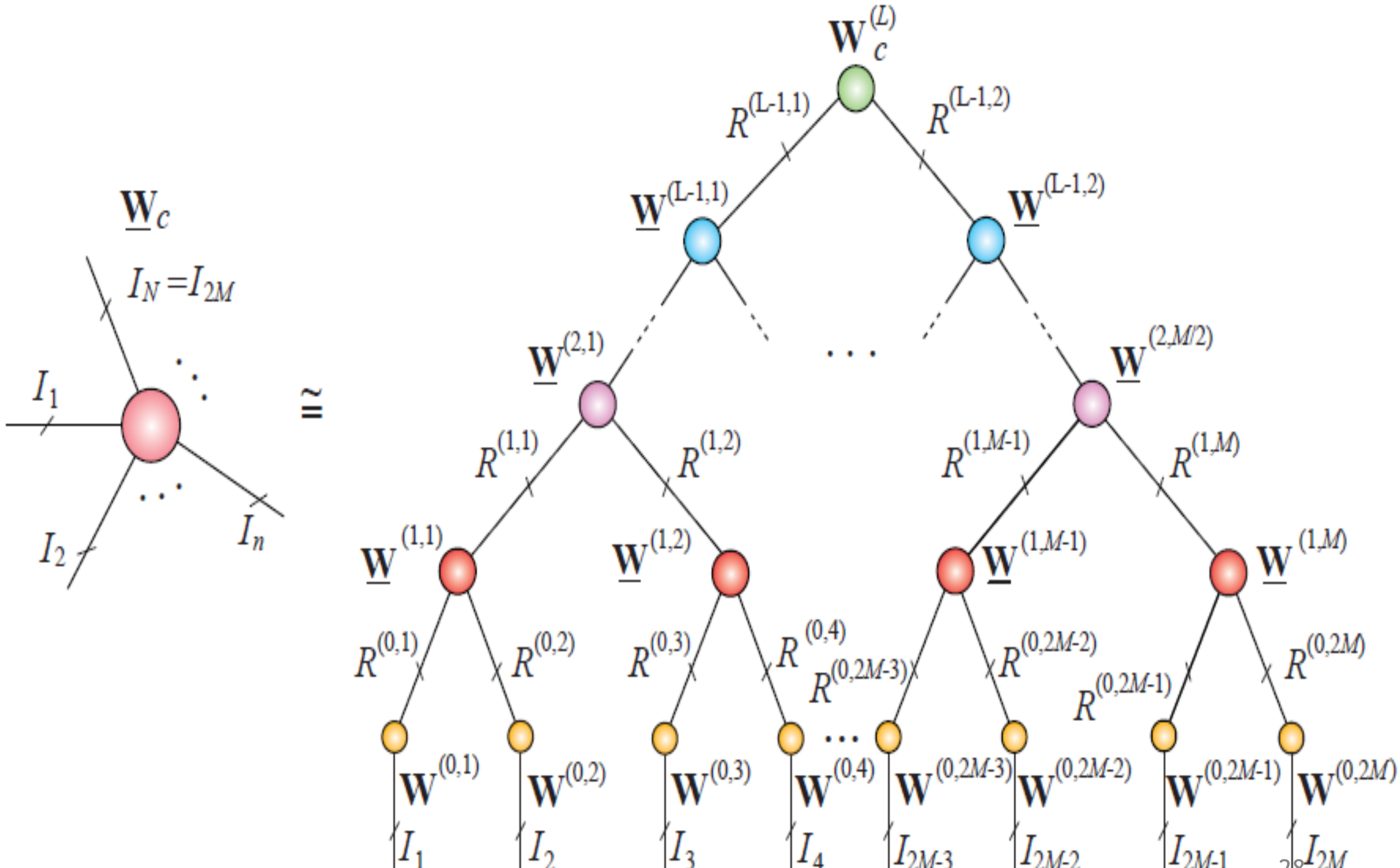


$$\mathbf{F} \in \mathbb{R}^{I \times I}, \text{ with rows } \mathbf{F}(i,:) = [f_{\theta_1}(\mathbf{x}^{(i)}), f_{\theta_2}(\mathbf{x}^{(i)}), \ldots, f_{\theta_I}(\mathbf{x}^{(i)})]$$

Simplified Hierarchical Tucker (Cohen - Shashua 2016/2017)



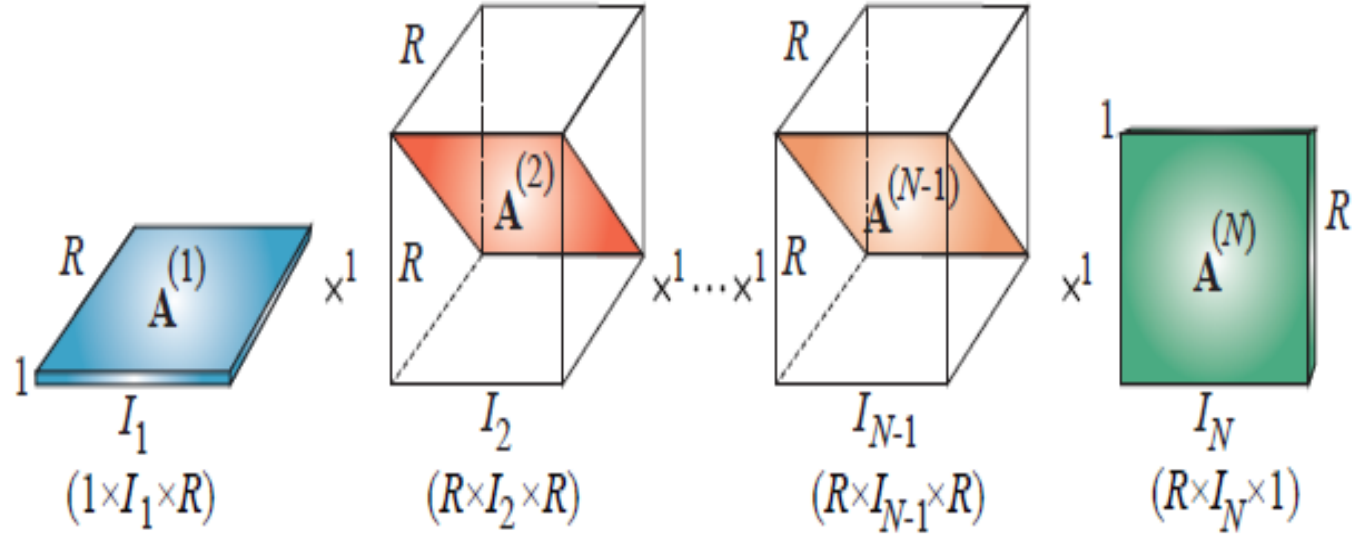
Standard HT



W_c
I_N = I_2M
I_1
I_2
I_n
≅
W_c^(L)
R^(L-1,1)
R^(L-1,2)
W^(L-1,1)
W^(L-1,2)
W^(2,1)
W^(2,M/2)
R^(1,1)
R^(1,2)
R^(1,M-1)
R^(1,M)
W^(1,1)
W^(1,2)
W^(1,M-1)
W^(1,M)
R^(0,1)
R^(0,2)
R^(0,3)
R^(0,4)
R^(0,2M-3)
R^(0,2M-2)
R^(0,2M-1)
R^(0,2M)
W^(0,1)
W^(0,2)
W^(0,3)
W^(0,4)
W^(0,2M-3)
W^(0,2M-2)
W^(0,2M-1)
W^(0,2M)
I_1
I_2
I_3
I_4
I_2M-3
I_2M-2
I_2M-1
I_2M

Links Between TT and CPD (Oseledets, 2011)

$$\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \circ \mathbf{a}_r^{(N)}$$

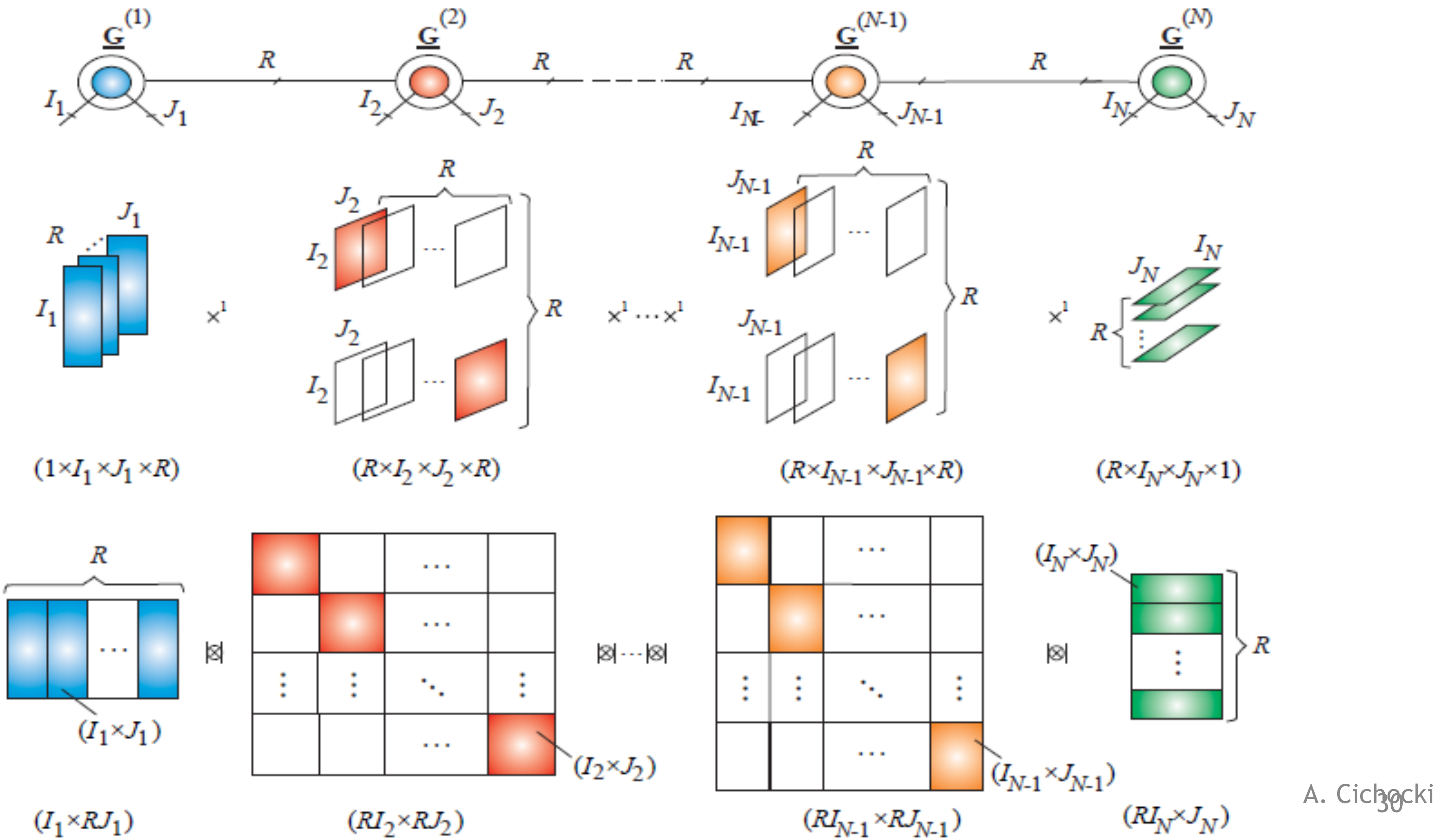


Links Between TT/MPO and CPD (Oseledets, 2011)

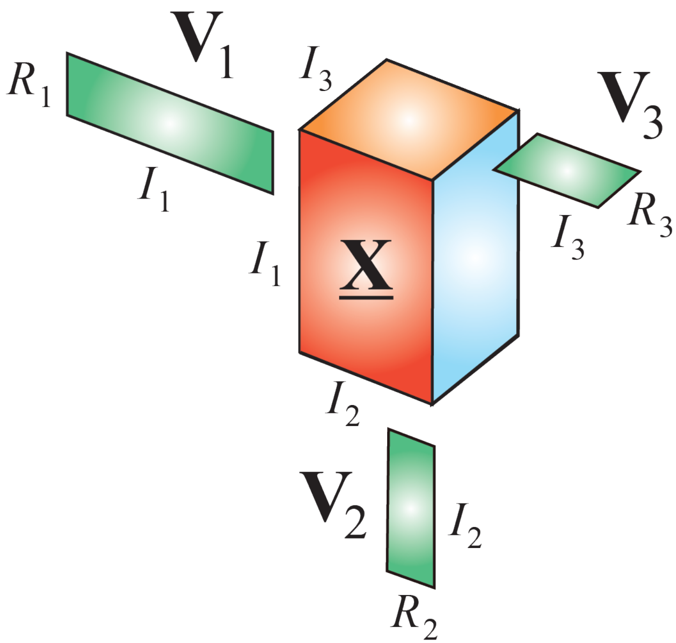
$$\underline{\mathbf{X}} = \sum_{r=1}^{R} (\mathbf{A}_r^{(1)} \circ \mathbf{A}_r^{(2)} \circ \cdots \circ \mathbf{A}_r^{(N)})$$

$$\mathbf{X} = \sum_{r=1}^{R} (\mathbf{A}_r^{(1)} \otimes_L \mathbf{A}_r^{(2)} \otimes_L \cdots \otimes_L \mathbf{A}_r^{(N)})$$

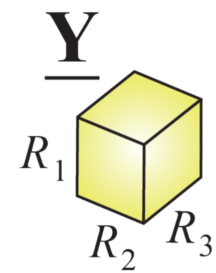
$$= \tilde{\mathbf{G}}^{(1)} |\otimes| \tilde{\mathbf{G}}^{(2)} |\otimes| \cdots |\otimes| \tilde{\mathbf{G}}^{(N)} \in \mathbb{R}^{I_1 \cdots I_N \times J_1 \cdots \times J_N}$$



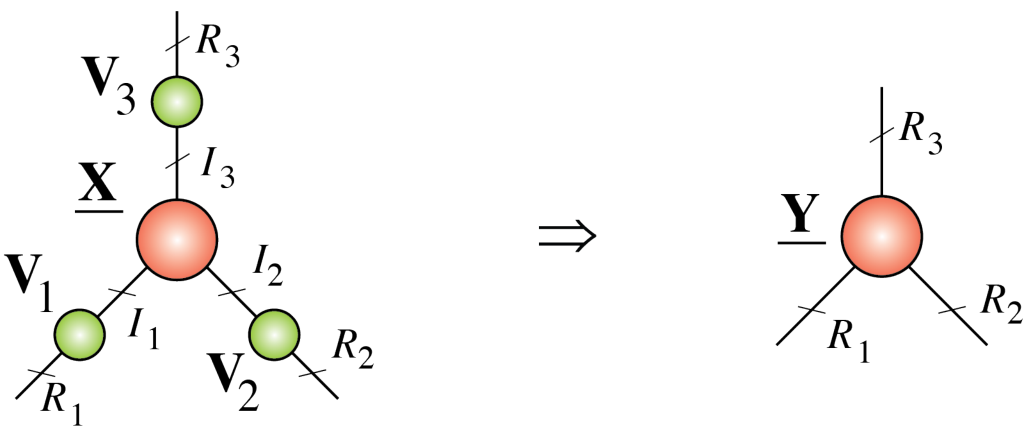
Tucker Models for Fully Connected Layers



=

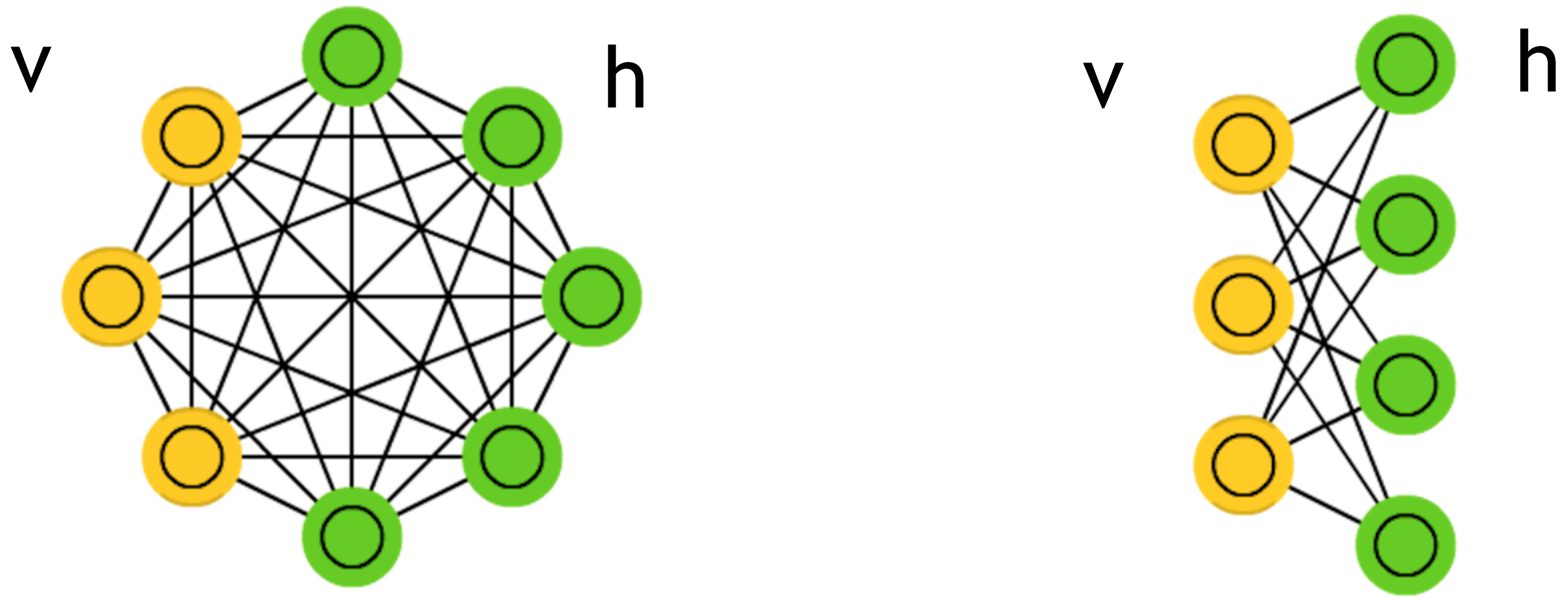


$$\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_1 \mathbf{V}_1 \times_2 \mathbf{V}_2 \times_3 \mathbf{V}_3$$

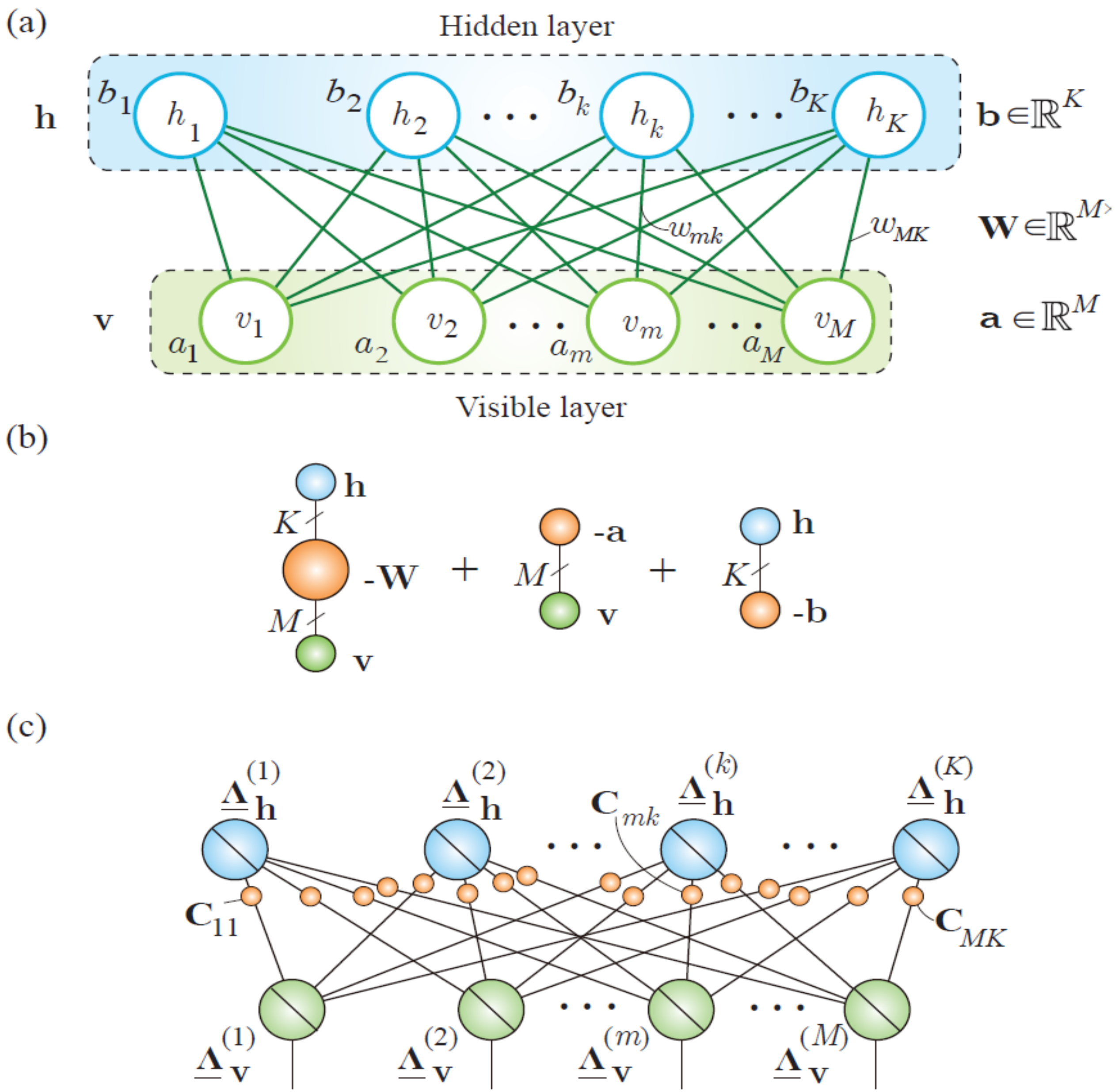


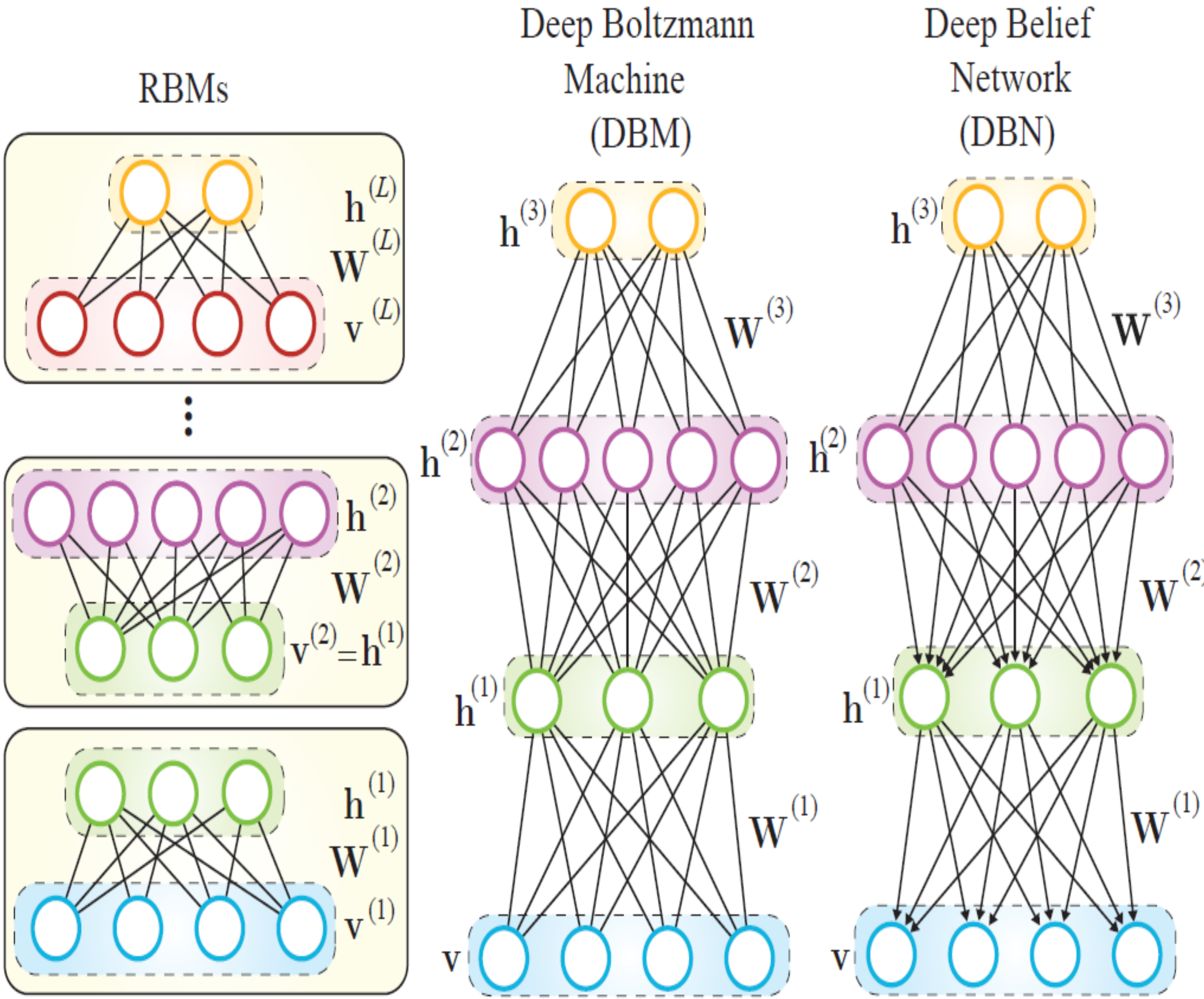
Boltzmann Machine
Intractable Problem Curse of Dimensionality

Visible layer   Hidden layer



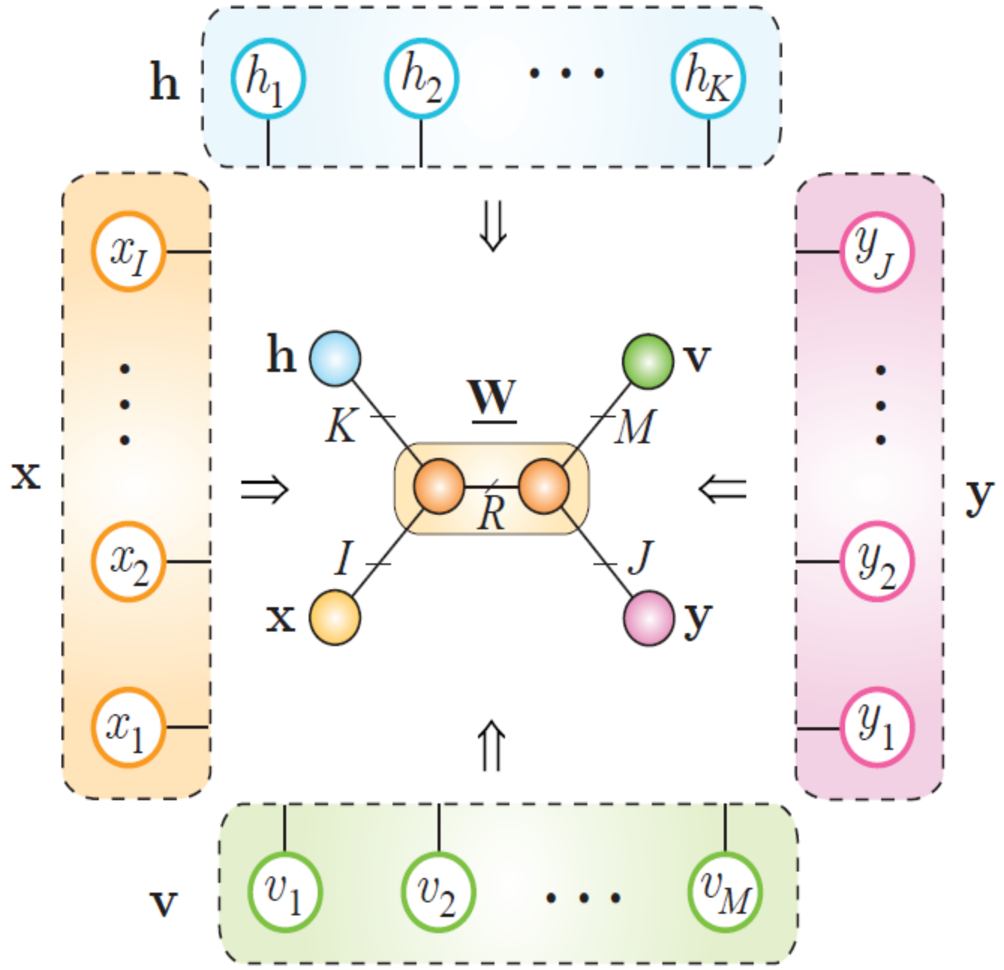
RBM (Restricted Boltzmann Machine)
Equivalence RBM and TN



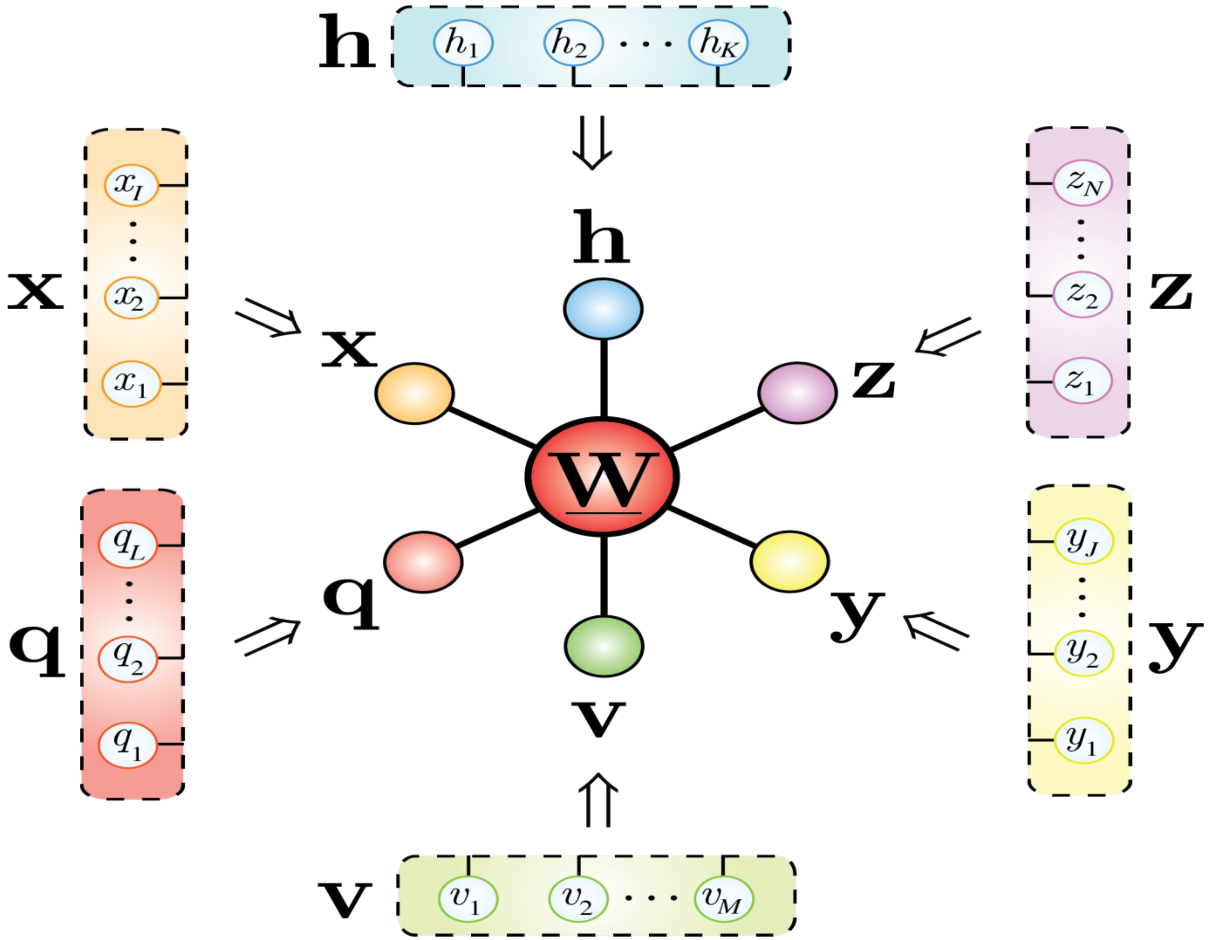

RBMs
Deep Boltzmann Machine (DBM)
Deep Belief Network (DBN)
h(L)
W(L)
v(L)
h(2)
W(2)
v(2)=h(1)
h(1)
W(1)
v(1)
h(3)
W(3)
h(2)
W(2)
h(1)
W(1)
v
h(3)
W(3)
h(2)
W(2)
h(1)
W(1)
v

A. Cichocki

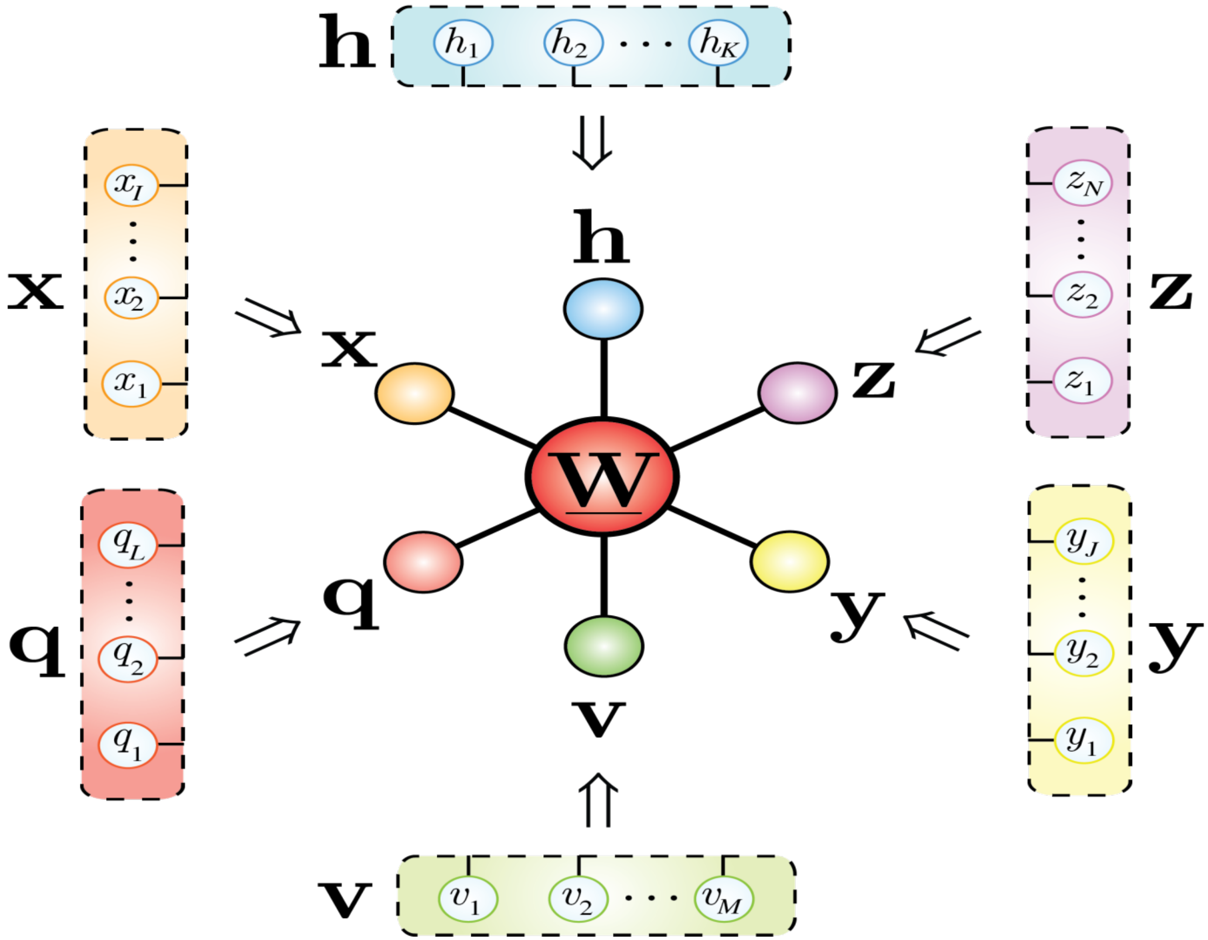
Multi-way RBMs (partially restricted)

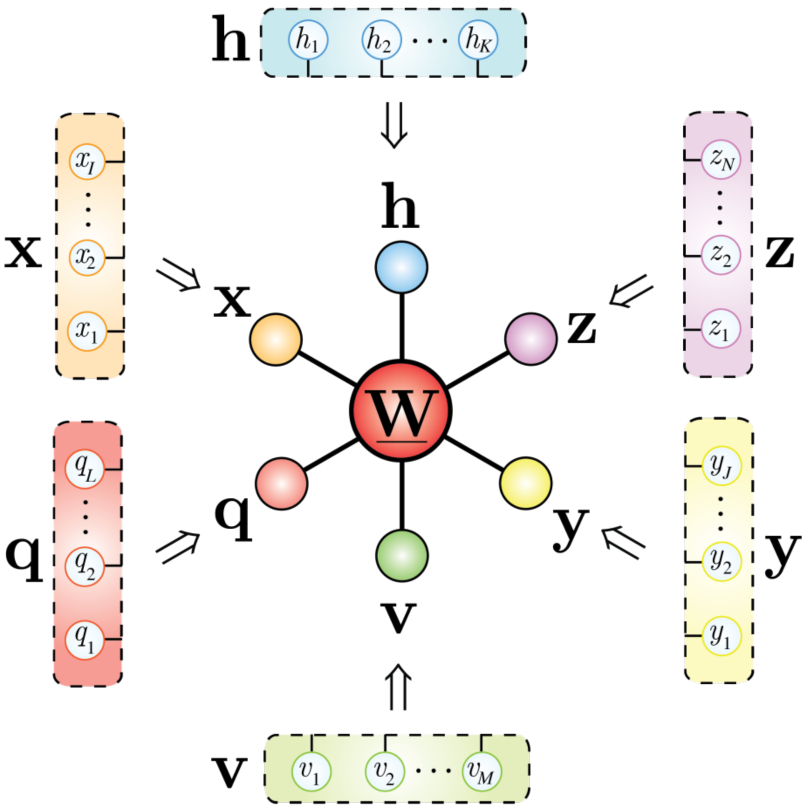


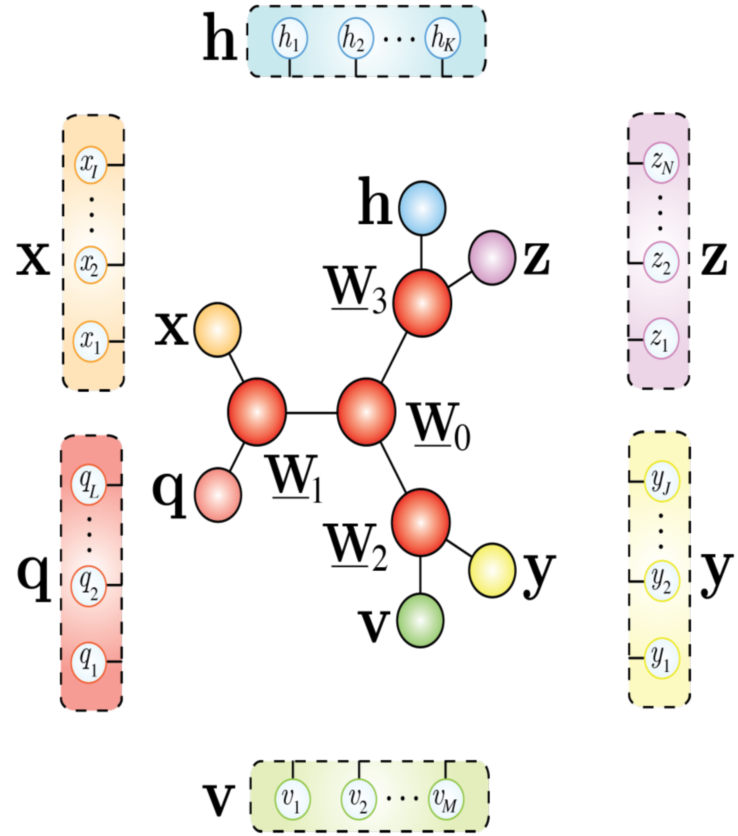
Multi-way Partially Restricted Boltzmann Machine (RBM)

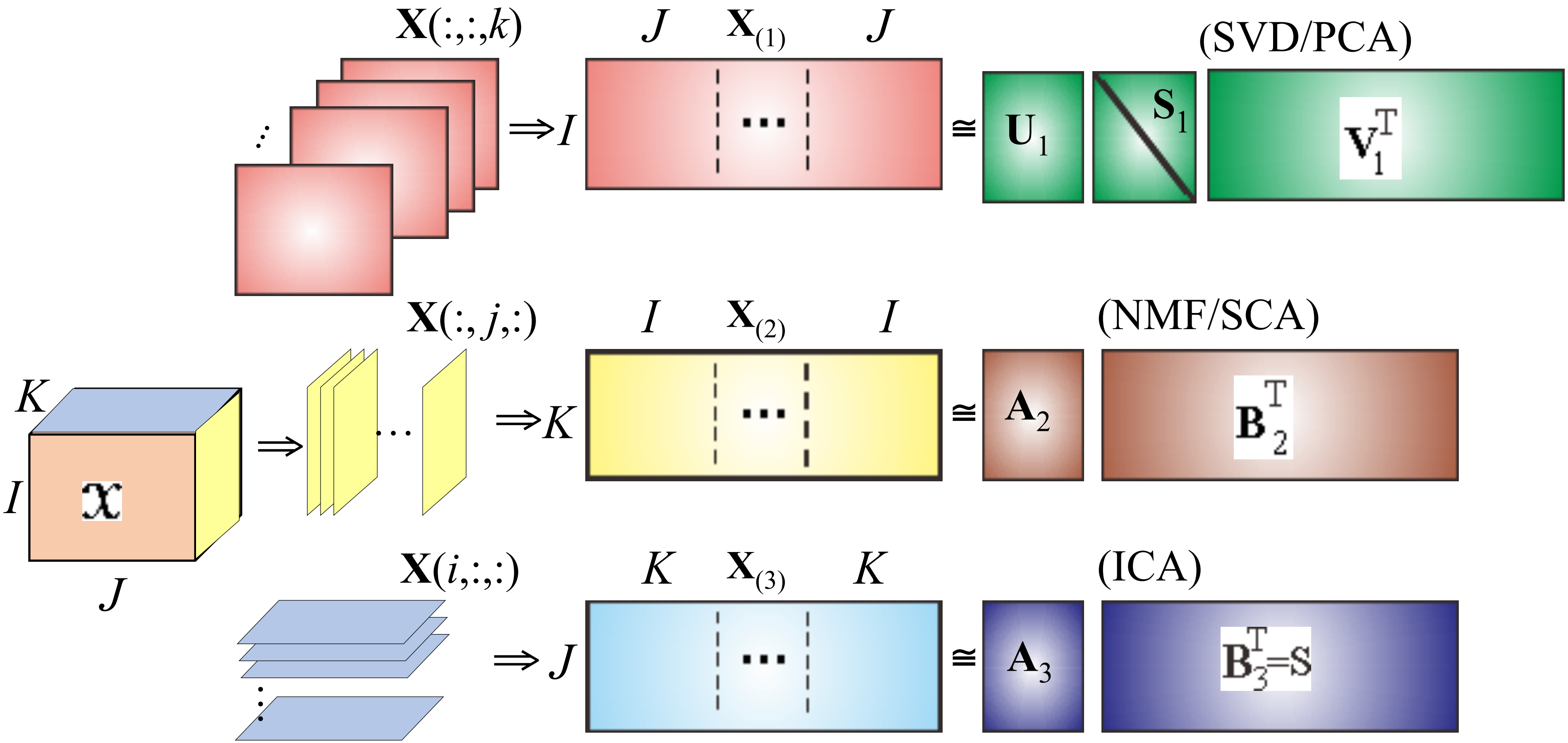


Multi-way Partially Restricted Boltzmann Machine (RBM)




h
h_1 h_2 ⋯ h_K
x
x_I
x_2
x_1
z
z_N
z_2
z_1
q
q_L
q_2
q_1
y
y_J
y_2
y_1
W
v
v_1 v_2 ⋯ v_M


h
h_1 h_2 ⋯ h_K
x
z
q
y
W_3
W_0
W_1
W_2
v
v_1 v_2 ⋯ v_M



X(:,:,k)
J
X(1)
J
(SVD/PCA)
I
U1
S1
V1T
X(:, j,:)
I
X(2)
I
(NMF/SCA)
K
K
A2
B2T
I
J
X(i,:,:)
K
X(3)
K
(ICA)
J
A3
B3T=S

Review papers

- Cichocki, A., Lee, N., Oseledets, I., Phan, A. H., Zhao, Q., & Mandic, D. P. (2016). Tensor networks for dimensionality reduction and large-scale optimization Part 1 Low-rank tensor decompositions. Foundations and Trends® in Machine Learning, 9(4-5), 249-429.
- Cichocki, A., Phan, A. H., Zhao, Q., Lee, N., Oseledets, I., Sugiyama, M., & Mandic, D. P. (2017). Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 2 Applications and Future Perspectives. Foundations and Trends® in Machine Learning, 9(6), 431-673.
- Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., & Phan, H. A. (2015). "Tensor decompositions for signal processing applications: From two-way to multiway component analysis». IEEE Signal Processing Magazine, 32(2), 145-163.