Hodge-Kodaira decomposition of evolving neural networks

Topics in Differential Geometry And its Discretizations

Jan 10, 2015

Keiji Miura and Takaaki Aoki

Table of contents

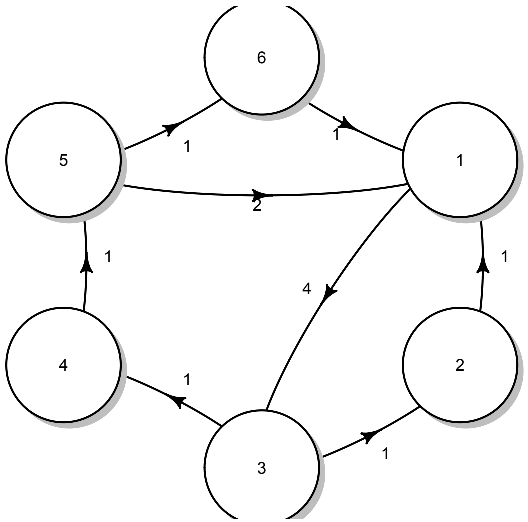
- Method: Hodge decomposition of a network
- Application: evolving neural networks

Hodge decomposition of a network

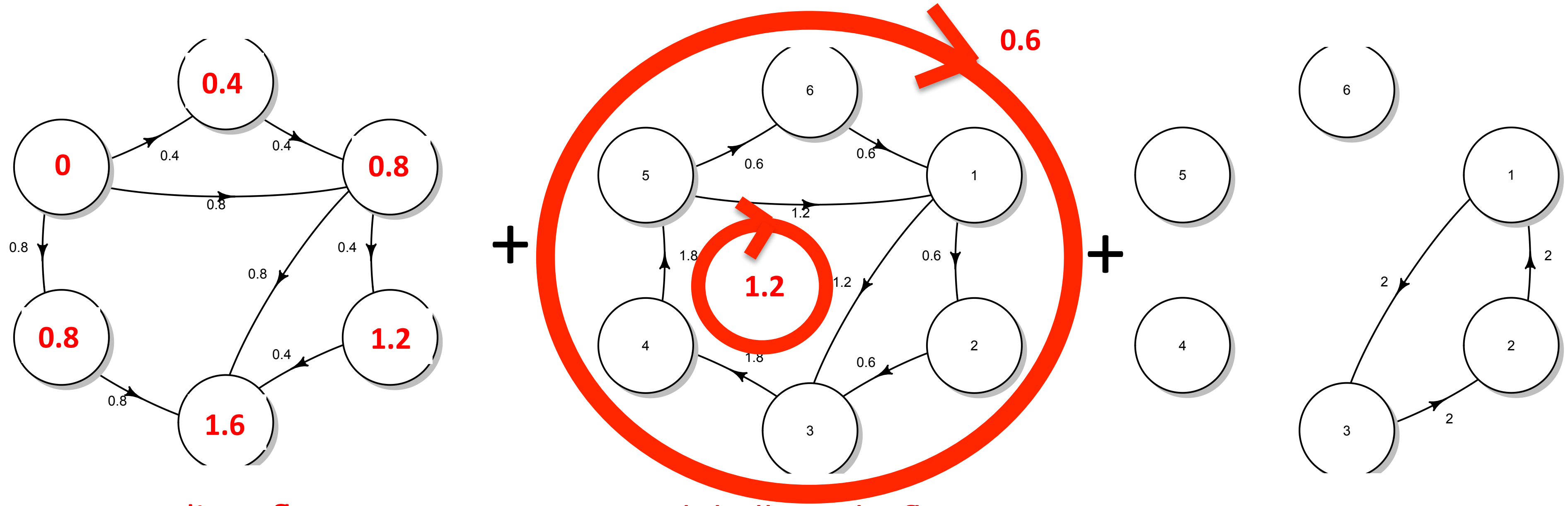
- Hodge-Kodaira decomposition of a network (such as neural networks or state transition diagrams) into three components detects global loops.
- Especially, the number of loops (1-χ), a graph invariant, can be informative.

  χ := #(node) - #(edge) + #(surface)

Jiang et al. (2011)



=

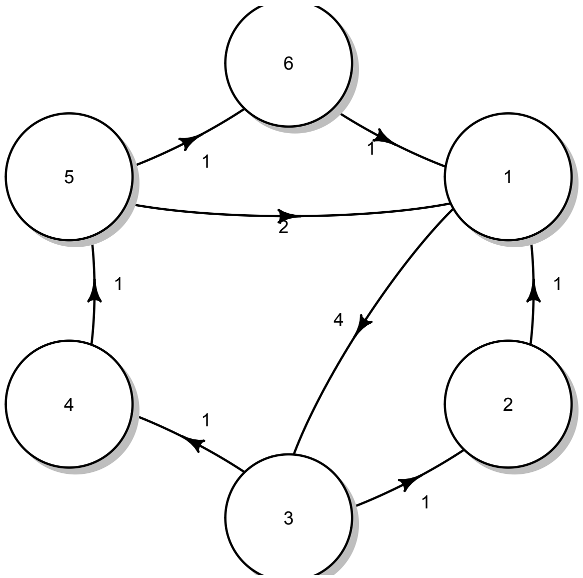


gradient flow   globally cyclic flow   locally cyclic flow

Possible applications

- Ranking (by potentials)
  - Layering neural networks
  - Animal hierarchy
  - Google rank alternative, ranking movies (Jiang et al '11)
- Loop detection (by cyclic flows)
  - Detection of recurrent connections
  - Inconsistency of ranking

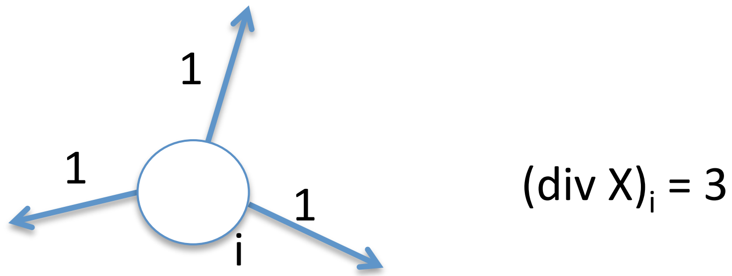
Matrix representation of a network



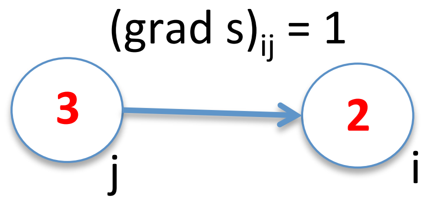
$$= \begin{pmatrix} 0 & -1 & 4 & 0 & -2 & -1 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 2 & 0 & 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

div, grad, curl for a graph

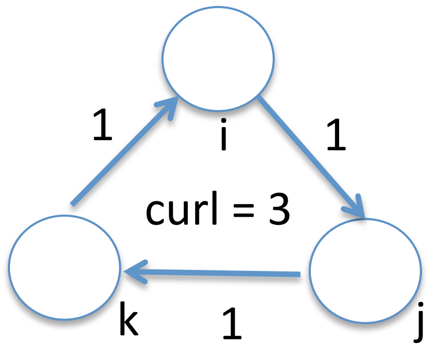
$$(\mathrm{div}\, X)(i) = \sum_j X_{ij}$$



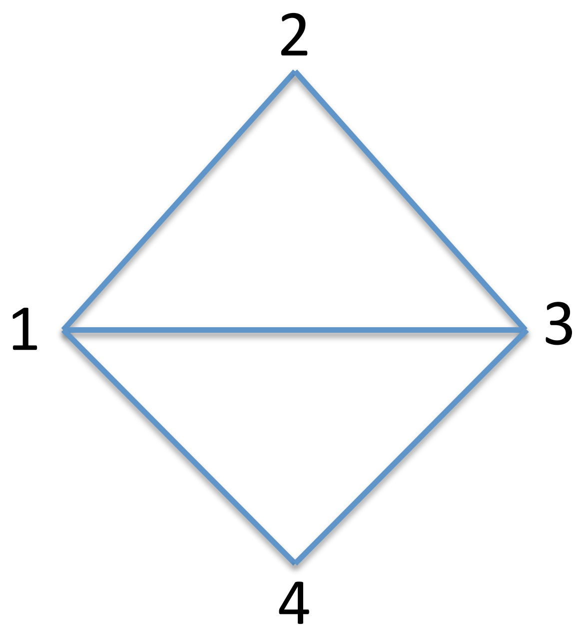
$$(\mathrm{grad}\, s)(i,j) = s_j - s_i$$



$$(\mathrm{curl}\, X)(i,j,k) = X_{ij} + X_{jk} + X_{ki}$$



Examples of operators



$$\begin{pmatrix} X_{12} \\ X_{13} \\ X_{14} \\ X_{23} \\ X_{34} \end{pmatrix} = \mathrm{grad}(\vec{s}) := \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}$$

$$\begin{pmatrix} C_{123} \\ C_{134} \end{pmatrix} = \mathrm{curl}(\vec{X}) := \begin{pmatrix} 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_{12} \\ X_{13} \\ X_{14} \\ X_{23} \\ X_{34} \end{pmatrix}$$

Essence of decomposition of any flows into 3 classes of flows

all possible potentials (functions on vertices) —grad→ all possible flows (functions on edges) —curl→ all possible curls (functions on “faces”)

Flows have 2 x 2 possibilities: (gradient or not) x (curl-free or not)
But, as “gradient is always curl-free”, one possibility disappears.

$$\mathrm{curl} \circ \mathrm{grad} = 0$$

Then, flows have only 3 classes:

- Gradient flow
- Non-gradient and curl-free flow (=: harmonic)
- Non-gradient and curl flow

Cf: continuous case (3d Euclidean)

$$\operatorname{grad}\psi = \left(\frac{\partial\psi}{\partial x_1}, \frac{\partial\psi}{\partial x_2}, \ldots, \frac{\partial\psi}{\partial x_n}\right) = \mathbf{e}_1\frac{\partial\psi}{\partial x_1} + \mathbf{e}_2\frac{\partial\psi}{\partial x_2} + \cdots + \mathbf{e}_n\frac{\partial\psi}{\partial x_n}$$

$$\operatorname{div}\mathbf{A} = \nabla\cdot\mathbf{A} = \frac{\partial A_x}{\partial x} + \frac{\partial A_y}{\partial y} + \frac{\partial A_z}{\partial z}$$

$$\operatorname{rot}\mathbf{A} = \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right)\mathbf{i} + \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right)\mathbf{j} + \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)\mathbf{k}$$

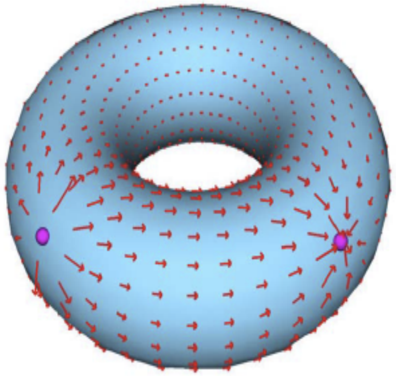
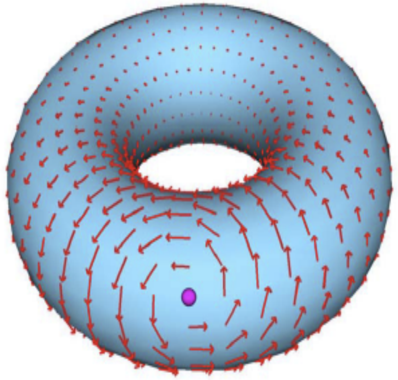
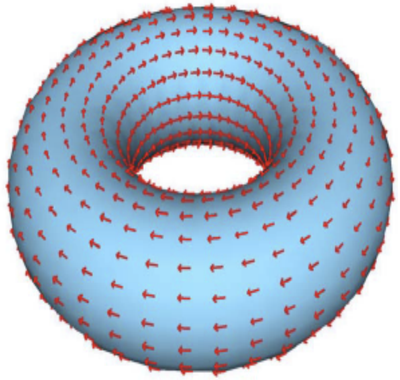
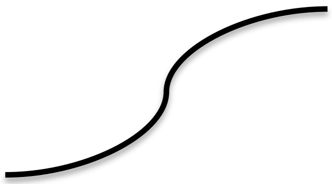
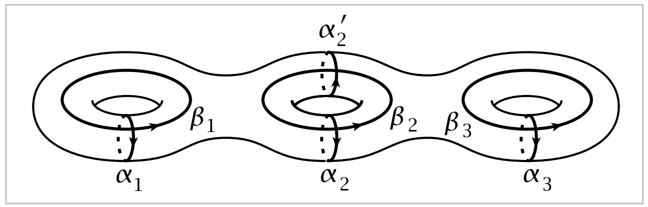
$$df = \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial z}dz$$

$$A = A_x dx + A_y dy + A_z dz$$

$$dA = \frac{\partial A_x}{\partial y}dy \wedge dx + \ldots = \left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right)dy \wedge dz + \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right)dz \wedge dx + \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)dx \wedge dy$$

Integral = inner product of flow & path

(K. Polthier, MAIPCV 2011)

		Gradient flow		Locally cyclic flow		Globally cyclic flow
Integrand (Cohomology)	=		+		+	
						Non-zero Integral
Path (Homology)	=	Not a boundary Not a loop	+	Boundary of a surface Loop	+	Not a boundary Loop
						

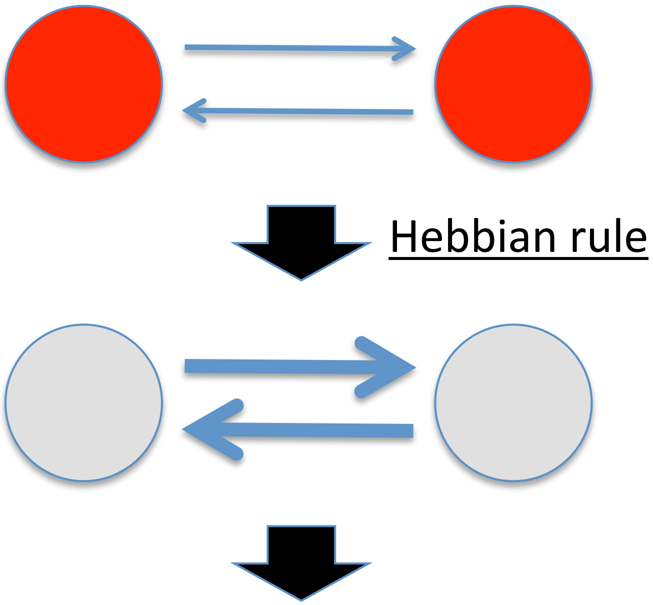
"Gradient is curl free" and "a boundary is a loop" guarantee decompositions into three components.
Cf. A slight change of a path by adding a local loop does not make difference in integral for a harmonic flow.

Table of contents

- Method: Hodge decomposition of a network
- Application: evolving neural networks

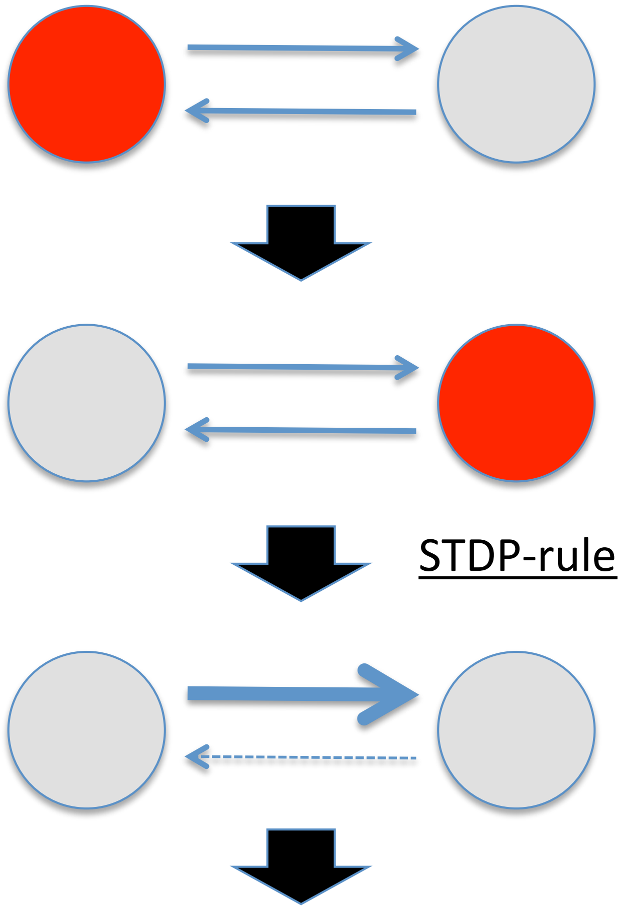
Biological learning rule generates loops

1. Hebbian learning rule



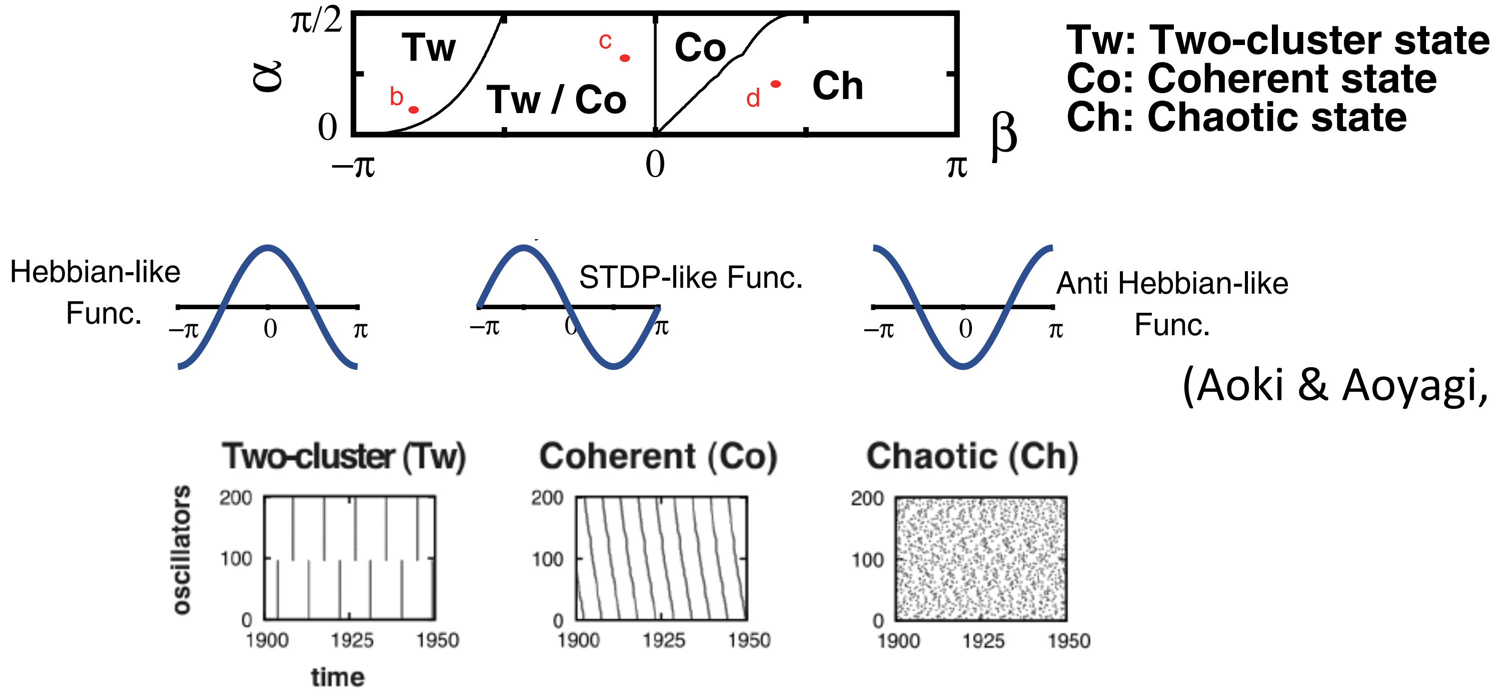
No loop (triangle or larger)

2. STDP learning rule



Many loops (and paths)

Aoki-Aoyagi Model: coupling strengths of a network change over time



(Aoki & Aoyagi, PRL 2009)

The goal is to subdivide the chaotic region in the bifurcation diagram.

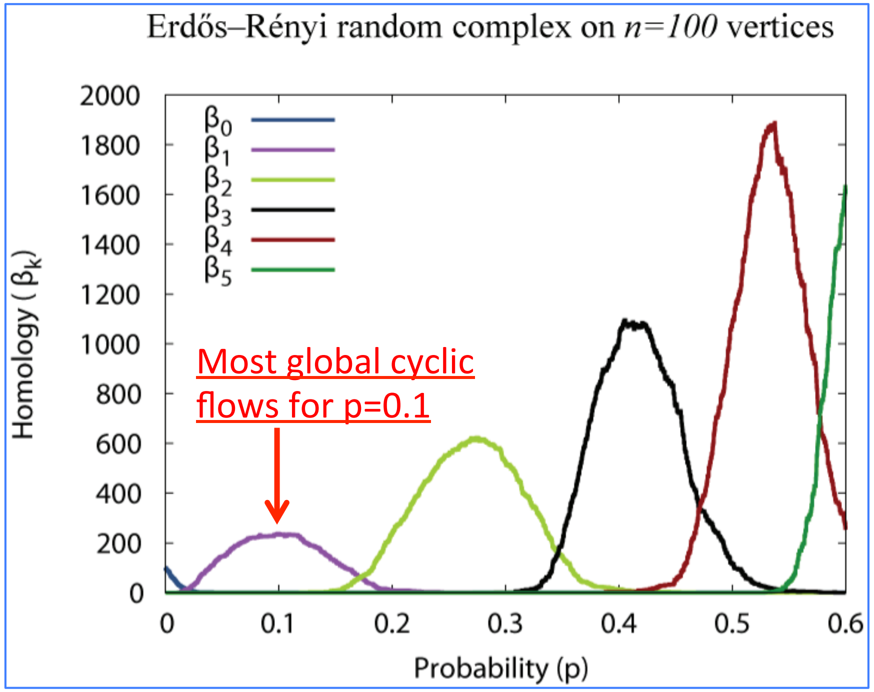
1. "Graph invariants", independent of labeling, are necessary as we start with random initial conditions.
2. STDP tends to make loops (Buonomano05,09,Magnasco09PRL).

→ #loops in Hodge decomposition can be useful?

Details of parameters

- N = 100 (#neuron, similar results for N=20-200)
- Regular random graph
- p=0.05-0.25 (p=0.1 optimal)

(Kahle and Meckes, 2010)



- Neurons as phase oscillators

$$\frac{d\phi_i}{dt} = \omega_i + \frac{1}{N}\sum_j \Gamma_{ij}(\phi_i - \phi_j),$$

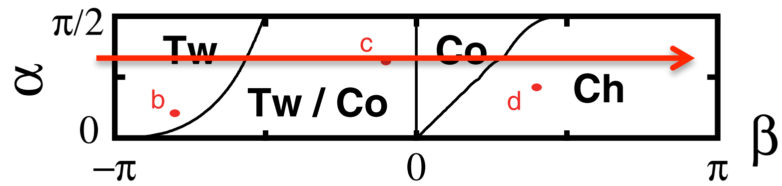
$$\frac{dk_{ij}}{dt} = \epsilon\Lambda_{ij}(\phi_i - \phi_j),$$

- Threshold = 0.05 or 0.2 < |coupling| for “existence” of edges
- α=0.1 or 0.3 (0.1 settled down quicker)

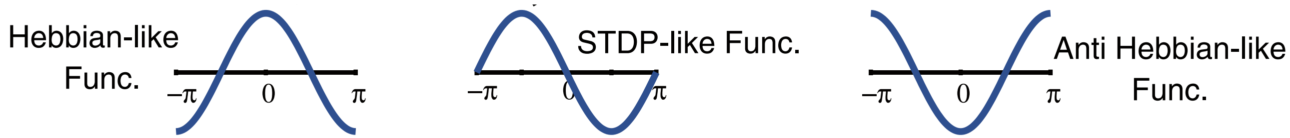
Aoki-Aoyagi Model: coupling strengths of a network change over time

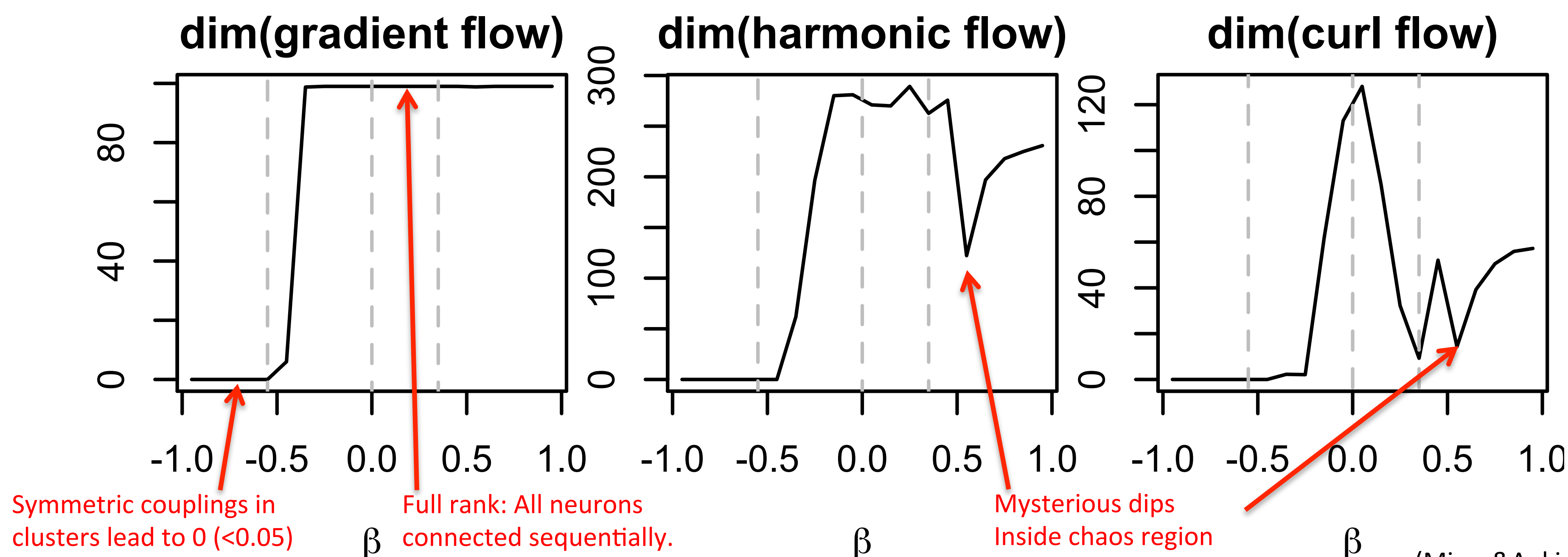
(Aoki & Aoyagi, PRL 2009)

α=0.3



Tw: Two-cluster state
Co: Coherent state
Ch: Chaotic state



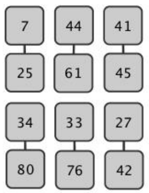


(Miura&Aoki, in press)

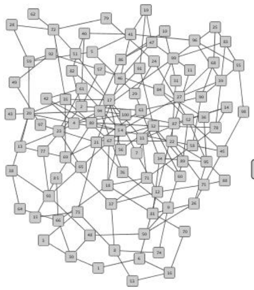
β=-0.95~-0.55
(#node,#edge)=(0,0)

none

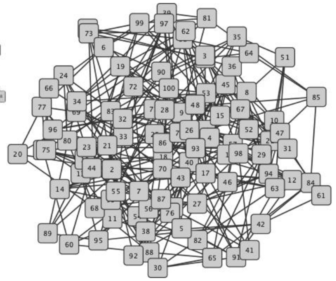
β=-0.45
(12,6)



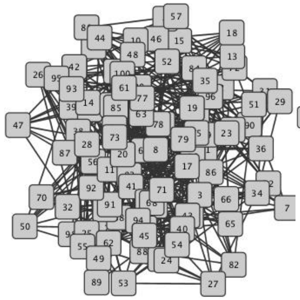
β=-0.35
(99,172)



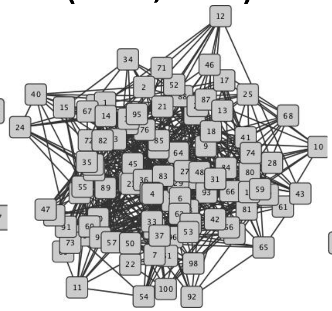
β=-0.25
(100,298)



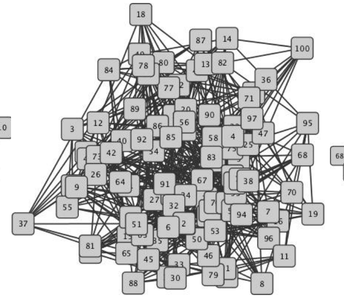
β=-0.15
(100,433)



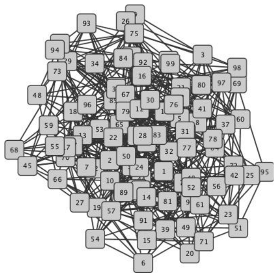
β=-0.05
(100,493)



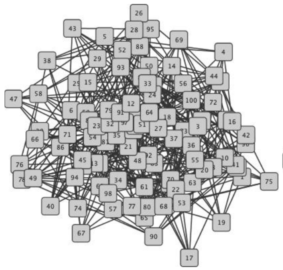
β=0.05
(100,498)



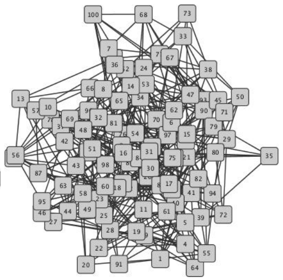
β=0.15
(100,454)



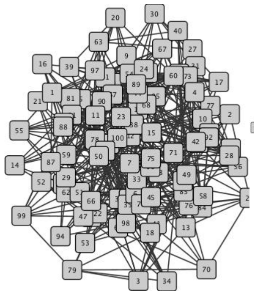
β=0.25
(100,419)



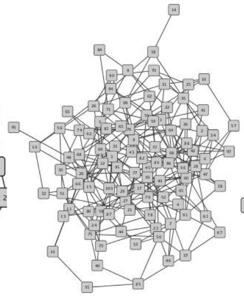
β=0.35
(100,364)



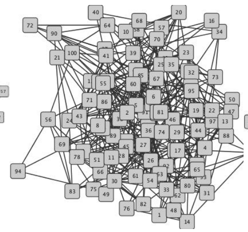
β=0.45
(100,411)



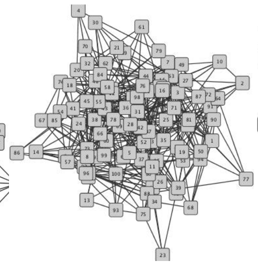
β=0.55
(100,245)



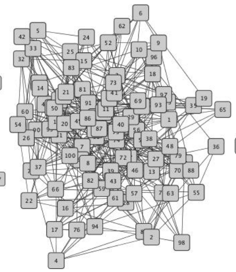
β=0.65
(100,340)



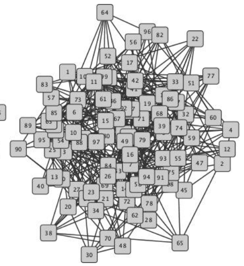
β=0.75
(100,358)



β=0.85
(100,385)



β=0.95
(100,389)



(α=0.3, threshold=0.05)

Summary

- We applied the Hodge-Kodaira decomposition to the evolving neural networks.
- A model with a STDP-rule, which tends to form paths coincident with causal firing orders, had the most loops.
- The dimension of each flow not only reflected the known bifurcation diagram but also detected the inhomogeneity inside the chaotic region.