Learning Using Dynamic Regime Identification and Synchronization

Nicolas Brodu

Presentation for WCCI06 / IJCNN

Outline

Main idea

Main idea: Synchronizing neuron behaviors

- By favoring connections as in Hebbian Learning.
- When the neuron behaviors are similar.

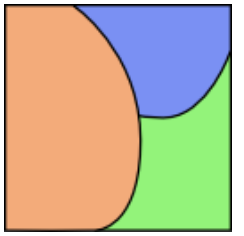
Problems:

- How to define “behavior”? What similarity is relevant?
- What to synchronize? Why?

Act locally, monitor globally:

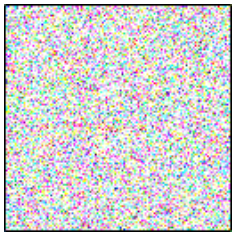
- Why should this setup even “learn” anything?
- How to avoid convergence to a global stasis?

Possible Global states



Totally ordered, convergent system

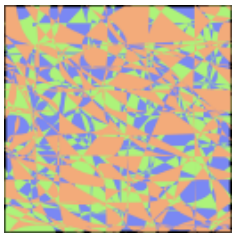
- Few final modes, well adapted to classification.
- Some initial information is “lost”, mapped N $\rightarrow$ 1.
- Hard to reverse, the system is not evolvable.



Totally Random

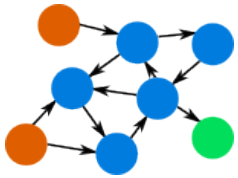
- Initial information is “lost”, robust property.

Critical line:



- NM-separation of initial information. [1]
- Much advertised concept of “edge of chaos”.
- Processing power, fading memory, and more. [2]

[1] T.Natschläger *et al.*, 2004.
[2] R. Legenstein, W.Maass, 2005

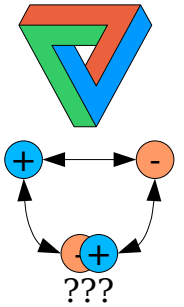


How to go critical

Synchronization

- Helps tighten efficient circuits, faster reaction.
- Tends to produce global order.

Frustration [3] (Incompatible constraints)

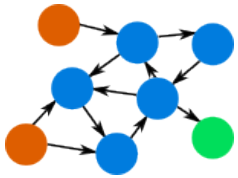


- Helps avoiding over-specialization.
- Tends to produce controlled chaos.

Of neuron local behaviors

- Time series, statistics, dynamical systems... [4]
- Get a synthetic ID for the neuron state.
- Synchronize and frustrate these IDs.

[3] H.Bersini, P.Sener, 2002
[4] C. Shalizi, 2006



Building learning rules

Choose a dynamical regime identifier

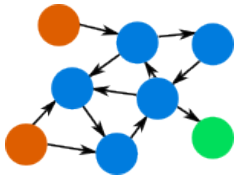
- What aspect of the neuron dynamics?
- What information does it carry?

Choose a significant target

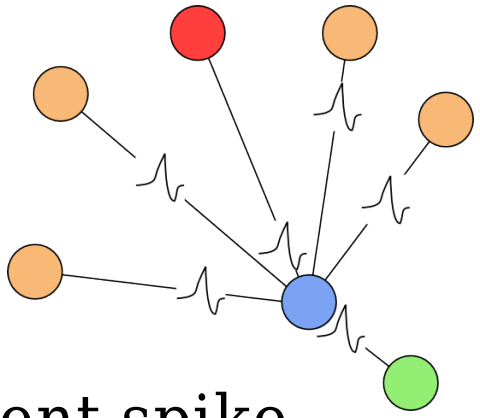
- Related to synchronization.
- Negative & positive learning cases.

Derive a learning rule

- Prevent global order and total randomness.
- Prevent over-specialization of connections.
- Maximal processing capabilities on a critical line?



Hebbian Learning



Spiking neural network version [5].

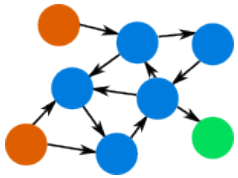
Observable:

- Relevant info: contribution to spiking.
- Δt difference between afferent and efferent spike times.
- Rationale: Favor the afferent neuron that triggers the efferent neuron spike.

Analysis:

- This is a kind of synchronization: goal is $0<\Delta t<\varepsilon$.
- Only one afferent node triggers the spike: frustration for the others (competitive learning).

[5] S. Song *et al.*, 2000



Validation method

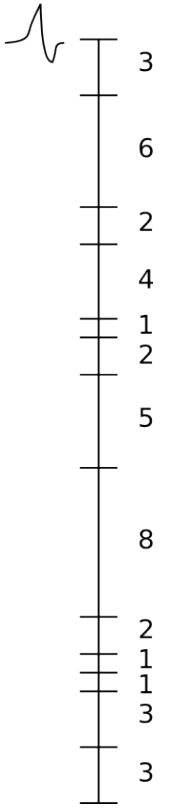
If the main idea is correct:

- The particular Hebbian Learning Δt choice is not the only possible observable.
- Any reasonable regime identifier would work.
- With an adequate learning rule.

Validation proposal:

- Find identifiers sufficiently different from Δt.
  - Another information is used, it cannot be equivalent.
  - Yet it should be local: a node or connectivity property.
- Derive learning rules.
- Observe the similitudes and differences.

A possible rule



Candidate observable: Neuron inter-spike time

- Not concerned by afferent/efferent times.
- Internal to a neuron, unlike Hebbian learning.
- But: Average is the activation frequency inverse, and averaging looses dynamical information.

Refinement to get part of the sequence info.:

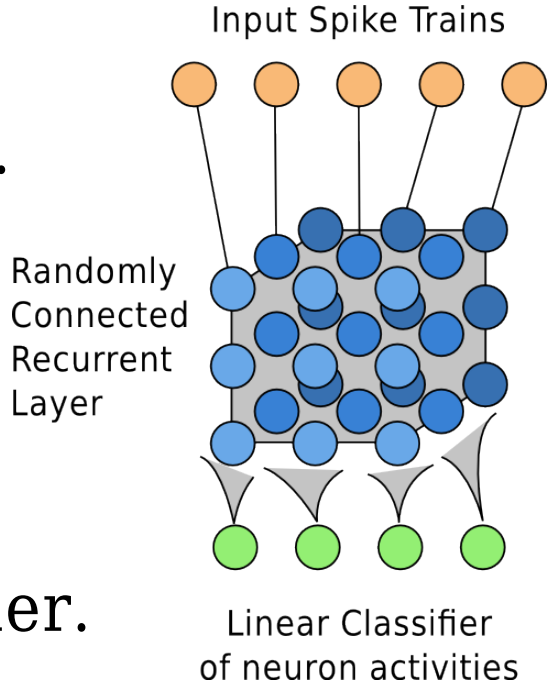
- Time/Frequency decomposition?
- Other statistics/time series methods?
- Choice to use Multifractal Analysis (DFA/Wavelet [6]):
  - Efficient [7], different enough from Hebbian Learning
  - Gives a synthetic ID to use for synchronization.

[6] P.Manimaran *et al.*, 2005
[7] N.Brodu, 2005

Application: Setup

Liquid State Machine [8] spiking neural network:

- Input layer receives spike trains.
- Randomized recurrent layer processes.
- Output layer receives activity signals.



Basic version:

- Recurrent layer does not learn.
- Output layer is actually a linear classifier.

Idea:

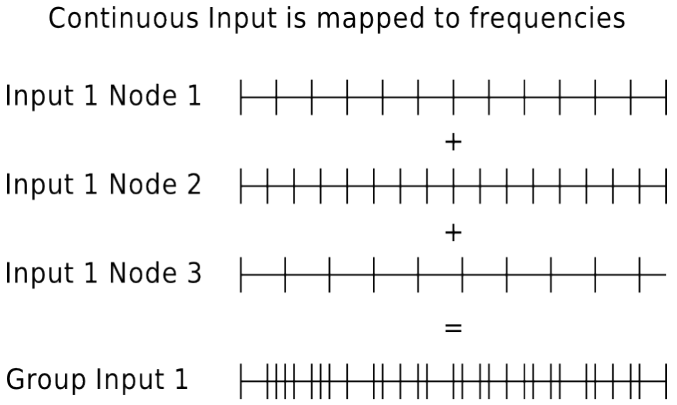
- Apply the previous rules to the recurrent layer.
- Compare performances with the base version.

[8] W.Maass *et al.*, 2003

Application: Data

Inputs:

- Use continuous data, both artificial, and real [9].
- Generate spike trains with frequency population coding.



Outputs:

- Classes for the input data.

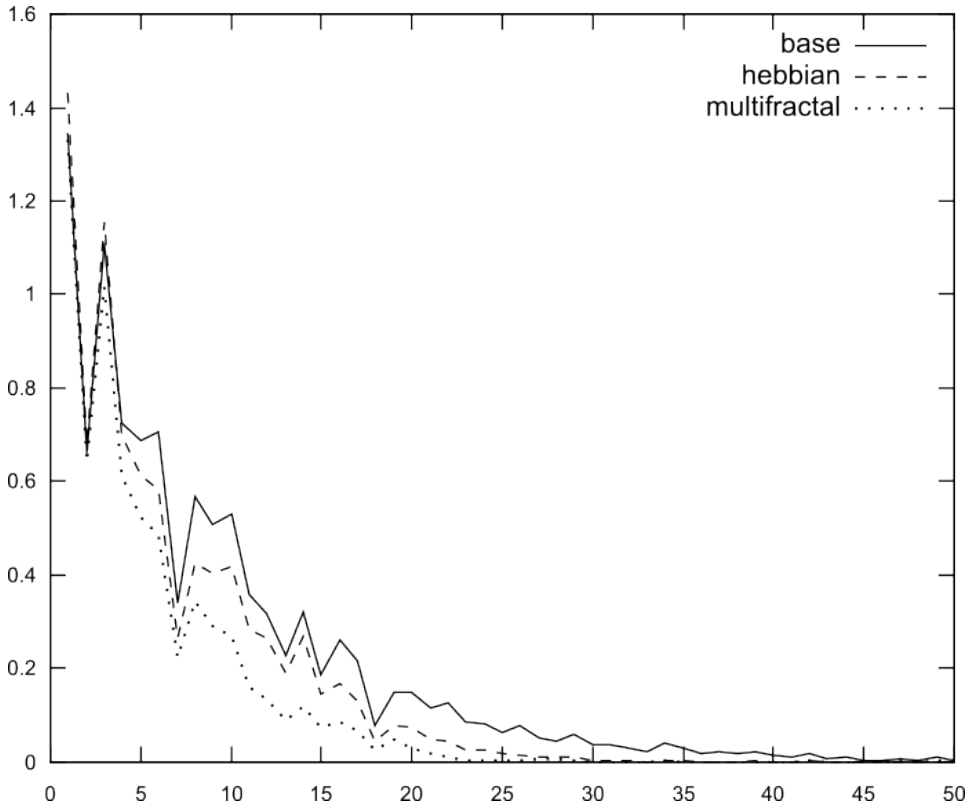
Learning measurement

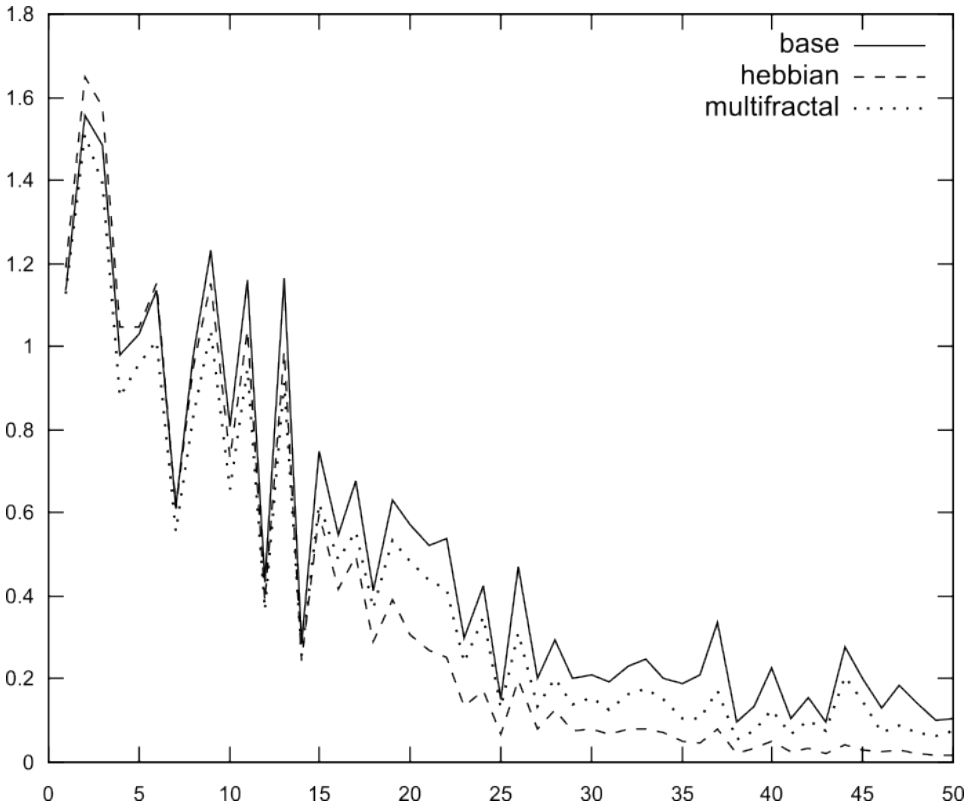
- Basic version classification performance.
- Hebbian & Multifractal classification performance.

Goal: Study similarities & differences

[9] L.Prechelt, 1994

Results: Training

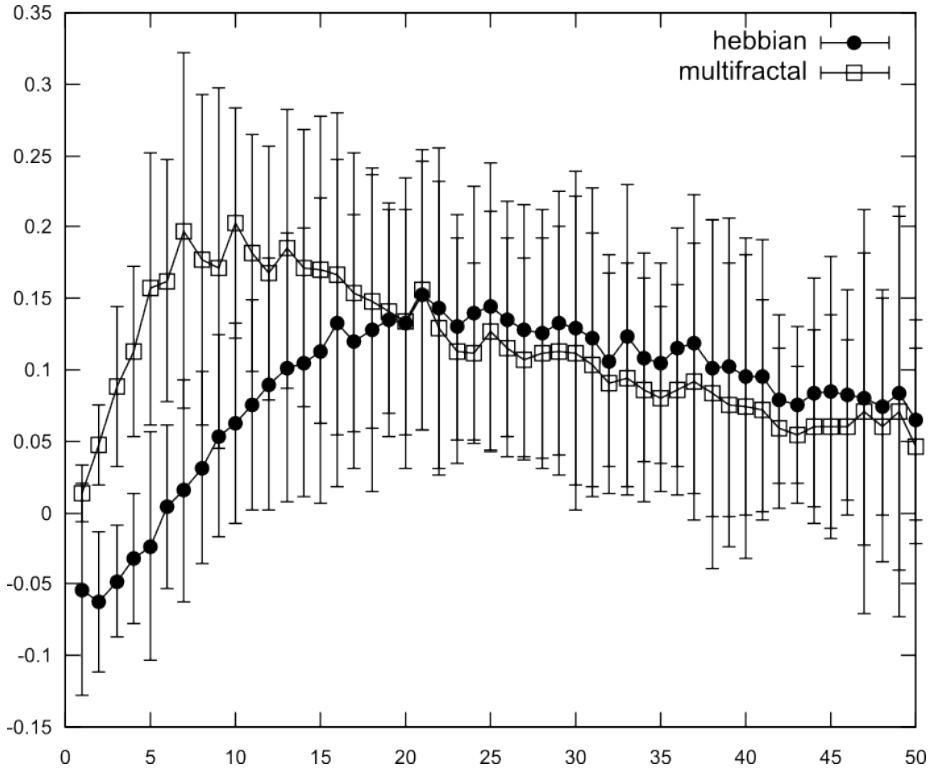


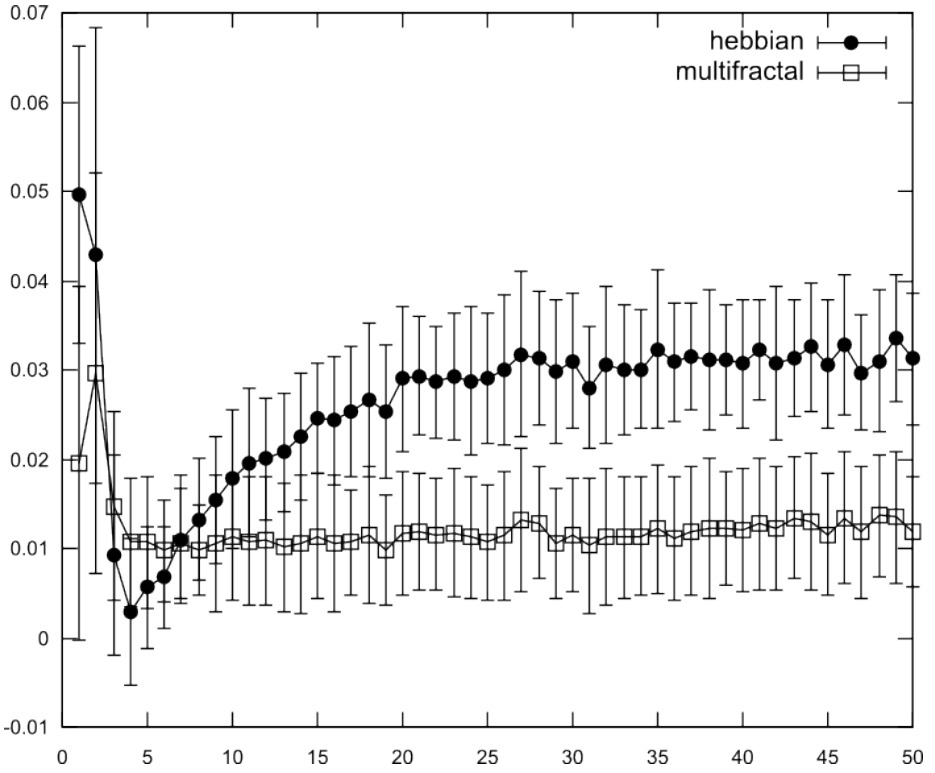


Two examples of learning error (error vs epochs).

Artificial data. Testing error is 0 in all cases.

Learning Comparison





Graphs: Improvement over the Basic version Mean & Dev.

Artificial (left) and Real data (right) (train error vs epoch).

Classification errors on real data, mean (dev): Base 6.20% (1.71%), Hebbian 5.70% (1.34%), MFA 5.73% (1.58%).

Conclusion

Retrospection:

- Theoretical method for generating learning rules.
- Application works and is illustrated by one rule.

Does this validate the theory generally?

- No, more rules and tests would be needed.
- Refining the theory would be interesting:
  - Better understanding of the role of synchronism and frustration in learning.
  - At least we have a practical approach how to do it.
  - Refined rules could lead to better learning.

All the code [10,11] is free-libre software: Use it!

[10] Amygdala project
[11] http://nicolas.brodu.free.fr

Full References

[1] T. Natschläger, N. Bertschinger, and R. Legenstein, “At the edge of chaos: real-time computations and self-organized criticality in recurrent neural networks”, in Proc. Advances in Neural Information Processing Systems, Dec. 2004.

[2] R. Legenstein and W. Maass, “What makes a dynamical system computationally powerful?” New Directions in Statistical Signal Processing: From Systems to Brain, Cambridge, MA: MIT Press, 2005.

[3] H.Bersini, P.Sener, “The connections between the frustrated chaos and the intermittency chaos in small Hopfield networks.”, in Neural Networks 15, 2002, pp1197-1024.

[4] Cosma Rohilla Shalizi (2006), “Methods and techniques of complex systems science: An overview”. Chapter 1, p. 33-114, Thomas S. Deisboeck and J. Yasha Kresh, Complex Systems Science in Biomedicine, 2006.

[5] S. Song, K. D. Miller, L. F. Abbott, “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity”, Nature Neuroscience. vol. 3, num. 9, pp 919-926, Sep. 2000.

[6] P. Manimaran, P. K. Panigrahi, and J. C. Parikh, “Wavelet analysis and scaling properties of time series”, Physical Review E, vol. 72, 046120, Oct. 2005.

[7] N. Brodu, “Real-time update of multi-fractal analysis on dynamic time series using incremental discrete wavelet transforms”, submitted for publication, Nov. 2005.

[8] W. Maass, T. Natschläger, and H. Markram “Computational models for generic cortical microcircuits” in Computational Neuroscience: A Comprehensive Approach, J. Feng, Ed. ch. 18, pp 575-605, 2003.

[10] Amygdala project, http://amygdala. sourceforge.net/

[11] http://nicolas.brodu.free.fr/en/recherche/publications/index.html