Binary vs. Analog Reservoir Computing: Connectivity and Performance

L. Büsing, B. Schrauwen and R. Legenstein

{lars,legi}@igi.tugraz.at
benjamin.schrauwen@elis.ugent.be
Introduction

Often two flavors of Reservoir Computing (RC) devices are considered:

- networks of **binary** (spiking) units: “LSMs”
  performance sensitive to details of connectivity
- networks of **analog** units: “ESNs”
  performance “robust” wrt. variations in connectivity

Systematical investigation via:

- Quantized ESNs:
  interpolation between binary and analog RC systems
- Influence of connectivity (parametrized by neuron in-degree) on computational performance

Büsing et. al. ()

Binary vs. Analog RC
1. Quantized ESNs and their Computational Performance
Definition (Quantized ESN)

- network state: \( x(t) = (x_1(t), \ldots, x_N(t)) \in (-1, 1)^N \) for \( N \) units

\[
x_i(t + 1) = (\psi_m \circ \tanh) \left( \sum_{j=1}^{N} w_{ij} x_j(t) + u(t) \right)
\]

- binary iid. input \( u(t) \in \{-1, 1\} \forall t \in \mathbb{Z} \) with \( p(u(t) = 1) = 1/2 \)

- quantization level \( m \in \mathbb{N} \) : determines admissible states per unit

\[
\psi_m : (-1, 1) \rightarrow S_m \subset (-1, 1)
\]

- state space \( S_m \) with cardinality \( 2^m \)

- all units have common in-degree \( K \in \mathbb{N} \)

- presynaptic units are randomly chosen

- all non-zero weights \( w_{ij} \sim \mathcal{N}(0, \sigma^2) \) are normally distributed: mean zero and standard deviation \( \sigma \)

*Investigate performance in \((\sigma, K)\)-parameter space*
Online computations

- approximate binary TI operator on $u(\cdot)$ with fading memory
- target output $y_T(t)$ at time $t$ function of the last $n$ input bits:
  \[
  y_T(t) := f_T(u(t-1), \ldots, u(t-n))
  \]
  \[
  f_T \in \{f|f : \{-1, 1\}^n \to \{-1, 1\}\}
  \]
- RC approach: train linear classifier $y(t) = \text{sign} \left( \sum_i a_i x_i(t) + b \right)$
- performance measure $p_{\text{exp}}(f_T)$: “memory capacity” for task $f_T$
  averaged over circuits (= weight matrices) $C$ for given $K$ and $\sigma$
  \[
  MC_{\tau}(f_T, C) := \frac{\text{cov} \left( y(t), y_T(t-\tau) \right)^2}{\text{var}(y(t)) \text{var}(y_T(t))} \in [0, 1]
  \]
  \[
  p_{\text{exp}}(f_T, K, \sigma) := \left\langle \sum_{\tau=0}^{\infty} MC_{\tau}(f_T, C) \right\rangle_{C|K,\sigma}
  \]
Performance $\rho_{\text{exp}}$ and the critical line

$m = 1$

$\Rightarrow$ Maximum performance at chaos-order phase transition

$\Rightarrow$ Maximum performance at chaos-order phase transition
Numerical results for $\rho_{\text{exp}}$

$m = 1$

$m = 3$

$m = 6$

Figure: $\rho_{\text{exp}}(\text{PAR}_5)$ for $\text{PAR}_n(u(t - 1), \ldots, u(t - n)) := \prod_{i=1}^{n} u(t - i)$

- small $m$:
  - maximum performance decreasing with $K$
  - region of good performance decreasing with $K$

- large $m$
  - maximum performance independent of $K$
  - region of good performance increasing with $K$
Further results for $p_{\text{exp}}$

Figure: $p_{\text{exp}}(\text{RAND}_5)$ averaged of 20 randomly chosen 5-bit tasks

Figure: $p_{\text{exp}}(\text{AND}_3)$ for 3-bit AND-task
2. Lyapunov Exponent and Rank Measure Analysis
Lyapunov exponent

Define “Lyapunov” exponent $\lambda$:

$$\lambda(K, \sigma) := \ln \left( \frac{\langle \delta_1(C) \rangle_{C|K,\sigma}}{\delta_0(m)} \right)$$

- $\delta_0(m)$: smallest admissible state perturbation for $m$
- $\delta_1(C)$: state distance after one time step
- quantifies average state separation caused by initial conditions

$\lambda < 0 \quad \lambda = 0 \quad \lambda > 0$

ordered   critical   chaotic

Lyapunov exponent

A. Quantization $m=1$ bit

B. Quantization $m=6$ bit

$\lambda = \log(\sigma) - \log(\sigma_0)$

$K=3$  
$K=12$  
$K=24$
Rank measure

Two aspects are important for high computational performance of a circuit $C$:

- **kernel rank** $r_k(C)$
  - quantifies “kernel quality”
  - high $r_k(C) = “rich”$ reservoir
  - $\approx$ memory about recent inputs

- **generalization rank** $r_g(C)$
  - low $r_k(C) = circuit C$ generalizes well
  - $r_k(C)$ linked to VC dimension
  - $\approx$ memory about remote inputs

Heuristic measure for computational performance:

$$p_{\text{rank}}(K, \sigma) = \left\langle r_k(C) - r_g(C) \right\rangle_{C|K,\sigma}$$

[R. Legenstein, W. Maass. Neural Networks 2007]
Rank measure

$\text{Rank measure}$

$m = 1$: increasing $K \rightarrow$ small region of efficient trade-off between memory about recent and remote inputs

$m = 6$: shift of $r_k$ and $r_g$
3. Mean-field Predictor for Computational Performance
NM-separation

NM: mean-field predictor for Computational Performance based on the separation ability of a binary network:

- $\text{NM} := s^* - f^* - i^*$
- large system limit ($N \to \infty$), annealed approximation
- pros:
  - prediction is quite accurate
  - relating performance and separation property
- cons:
  - computationally expensive
  - cannot easily be extended to quantization levels $m \neq 1$

[Bertschinger, Natschläger. Neu. Comp. 2003]
Single bit separation

\[ \text{Input} \quad u(t-1), u(t-2), \ldots, u(t-k), \ldots \]

\[ \mathbf{x}^1(t) \]

\[ \uparrow \]

\[ \mathbf{x}^2(t) \]

\[ \text{Input} \quad u(t-1), u(t-2), \ldots, -u(t-k), \ldots \]

\[ d(k) := \frac{1}{N} \left\langle \| \mathbf{x}^1(t) - \mathbf{x}^2(t) \|_1 \right\rangle_u \]
Novel mean-field predictor

Define single bit separation of bit $k$:

$$d(k) := \frac{1}{N} \langle \|x^1(t) - x^2(t)\|_1 \rangle_u$$

$$u_2(t - i) = +u_1(t - i) \quad \forall i \in \mathbb{N}\{k\}$$

$$u_2(t - k) = -u_1(t - k)$$

heuristic performance predictor $p_\infty$:

$$p_\infty := \max\{d(2) - d(\infty), 0\}$$

- $d(2)$: input separation on short (=relevant) time-scales
- $d(\infty)$: input separation on long (=irrelevant) time-scales
- efficient numerical evaluation possible for: $N \to \infty$ and annealed approximation
Novel mean-field predictor

\( m = 1 \)

\( m = 3 \)

\( m = 6 \)
Contributions to $\rho_{\infty}$

- $m = 1$: $\max_{\sigma} \{d(2) - d(\infty)\}$ decreases with increasing $K$
- $m = 6$: constant shift in $d(2)$ and $d(\infty)$
Input separation & memory function

single bit separation is an upper bound for the memory function $m(k)$:

$$m(k) \leq \frac{1}{4} \| C^{-1} \|_2 \cdot d(k)^2$$

- $C_{ij} = \langle x_i(t)x_j(t) \rangle$, $C^{-1}$ pseudo inverse of $C$
- bound also holds for analog ESNs
- for globally contracting analog ESNs:

$$m(k) = O(\exp(-2k\gamma)) \quad \text{for } \gamma > 0$$
4. Summary and Outlook
Quantized ESNs with quantization level $m$

- for small $m$: in-degree $K$ has large influence on computational capabilities $\rightarrow$ LSMs
- for large $m$: in-degree $K$ has little influence on computational capabilities $\rightarrow$ ESNs

- effect can be linked to:
  Lyapunov exponent, rank measure and separation property
Open questions

- Different connectivity graphs with the same in-degree show similar behavior. Is the in-degree the only quantity of interest?
- Can the observed effect explain the difficulties to tune a spiking network to a decent working regime?
- Why do cortical (spiking) neurons have a high in-degree?
Joint work with:

Benjamin Schrauwen

Robert Legenstein
Kernel rank

Classification task with two classes on \( l \) inputs \( u_1(\cdot), \ldots, u_l(\cdot) \)

\[
r_k(C) := \text{rank} \begin{pmatrix} x[u_1]_1(t) & \cdots & x[u_1]_N(t) \\ \vdots & \ddots & \vdots \\ x[u_l]_1(t) & \cdots & x[u_l]_N(t) \end{pmatrix}
\]

- \( x[u_i](t) = (x[u_i]_1(t), \ldots, x[u_i]_N(t)) \)
  state vector after applying input \( u_i(\cdot) \)
- if \( r_k(C) = l \): all binary classification tasks can be performed
- in general: \( r_k(C) \) “degree of freedom” for classification task
Generalization rank

Classification task with two classes on \( l \) inputs \( u_1(\cdot), \ldots, u_l(\cdot) \)

\[
    r_k(C) := \langle \text{rank} \begin{pmatrix}
    x[\tilde{u}_{i,1}]_1(t) & \cdots & x[\tilde{u}_{i,1}]_N(t) \\
    \vdots & \ddots & \vdots \\
    x[\tilde{u}_{i,o}]_1(t) & \cdots & x[\tilde{u}_{i,o}]_N(t)
    \end{pmatrix} \rangle_i
\]

- \( \tilde{u}_{i,j}(\cdot) \) for \( j \in \{1, \ldots, o\} \): noisy variations of \( u_1(\cdot) \)
  i.e. \( \tilde{u}_{i,j}(\cdot) \) are in the same class as \( u_i(\cdot) \)
- if \( r_g(C) \) small: network generalizes well
- \( r_g(C) \) linked to VC-dimension