Contents

1 Introduction

2 Theoretical background

2.1 General notion of artificial neural networks

Artificial neural networks are getting more and more attention over the last decades as advances in raw computational power make it possible to implement these computationally heavy algorithms as well as recent achievements in better than human performance on certain tasks. In the next paragraphs we provide basic theoretical concepts behind artificial neural networks.

2.1.1 Neural network model

Haykin [2] defines a neural network in the following way:

Definition 1 *A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural property for storing experiential knowledge and making it available for use. It resembles the brain in two respects:*

1. *Knowledge is acquired by the network from its environment through a learning process.*

2. *Interneuron connection strengths, known as synaptic weights, are used to store acquired knowledge.*

The basic information-processing unit is called a *neuron*. In mathematical terms the neuron k of a system consisting of N neurons, can be written as a pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{1}$$

$$y_k = \varphi(u_k + b_k) \tag{2}$$

where $x_1, x_2, \dots, x_m$ are the *input signals*, $w_{k1}, w_{k2}, \dots, w_{km}$ are *synaptic weights*, u_k is called *linear combiner output* due to the input signals, b_k is called *bias*, $\varphi(\cdot)$ is the *activation function*, and y_k is the output signal of the neuron.

There are two basic activation functions:

1. *Threshold function*

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \tag{3}$$

2. *Sigmoid function*

- logistic function

$$\varphi(v) = \frac{1}{1 + e^{-av}} \tag{4}$$

- hyperbolic tangent function

$$\varphi(v) = \tanh(v). \tag{5}$$

2.1.2 Neural network as an approximation

One way to look at artificial neural networks is as a mapping

$$f : X \to Y$$

where X is the set of inputs and Y is the set of outputs. In the language of statistics we are dealing with nonparametric statistical inference. This view is strongly supported by an important theoretical result proved by Hornik [5], namely the universal approximation theorem.

Theorem 1 *Let*

$$\mathfrak{N}_k^{(n)}(\varphi) = \left\{ y : \mathbb{R}^k \to \mathbb{R} \mid y(x) = \sum_{i=1}^{n} \beta_i \varphi(\sum_{j=1}^{m} w_{ij} x_j - b_i) \right\}$$

denote a set of all functions implemented by a network with n hidden units and one output unit. If φ is continuous, bounded and nonconstant, then $\mathfrak{N}_k^{(n)}$ is dense in $C(X)$ for all compact subsets X of $\mathbb{R}^k$ ($C(X)$ is the space of all continuous functions on X).

Here we can observe some similarities to the Stone-Weierstrass theorem (it implies that the set of all polynomials of X is dense in $C(X)$). The process of estimation of the synaptical weights w_{ij} for $i = 1, \ldots, n$ and $j = 1, \ldots, m$ is called *learning*.

2.1.3 Classes of neural networks

The neurons of a neural network can be organized into many different structures. Many have been already developed and successfully applied to different tasks. The structure is called *network architecture* and in general there are two fundamentally different classes of network architectures:

1. **Feedforward networks**

   The structure of neurons is organized in layers. There is an *input layer*, *hidden layer* and *output layer*. The synaptical weights are only between neurons of two concurrent layers and only in one direction. In other words the output signals from one layer serve as input signals to the neurons in the second layer. We say that the network is *fully connected* if every unit in each layer of the network is connected to every unit in the next layer. Otherwise we say that the network is *partially connected*. In general there can be multiple hidden layers. Theorem 1 deals with a feedforward network with a single hidden layer. The standard learning algorithm for feedforward networks is called *backpropagation of error*. Basically it is a form of gradient based optimisation method.

2. **Recurrent networks**

   A recurrent neural network has at least one *feedback loop*. Feedback as is used in dynamical systems, that is when the output influences the input of an element of the system. In the setting of neural network let us consider a linear operators A and B, and an input-output relationship between two neurons

   $$y_k(n) = A[x_j^{'}(n)]$$

   and

   $$x_j^{'}(n) = x_j(n) + B[y_k(n)].$$

   Modifying these equations we get

   $$y_k(n) = \frac{A}{1-AB}[x_j(n)].$$

   We call the term $\frac{A}{1-AB}$ a *closed-loop operator* of the system, and AB the *open-loop operator*. Generally the existence of feedback loops in the neural network make the process of learning very difficult.

2.2 Echo state network

Echo state networks (ESNs) belong to the class of recurrent neural networks. This type of architecture was first proposed by Jaeger [3]. In the next paragraphs we will provide the structure and some properties of an echo state network.

2.2.1 Architecture

The architecture consists of input layer, reservoir and output layer. The reservoir is a fully connected recurrent hidden layer. The update equations are given by

$$\bar{\mathbf{x}}(t) = \tanh(\mathbf{W}^{in}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)) \tag{6}$$

$$\mathbf{x}(t) = (1-\alpha)\mathbf{x}(t-1) + \alpha\bar{\mathbf{x}}(t) \tag{7}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N_x}$ is vector of reservoir neuron activations and $\bar{\mathbf{x}}(t) \in \mathbb{R}^{N_x}$ is its update at time step t, $\tanh(\cdot)$ is the activation function, $\mathbf{W}^{in} \in \mathbb{R}^{N_x \times (1+N_u)}$ is the input weight matrix, $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$ is the recurrent weight matrix, and $\alpha \in (0, 1]$ is the leaking rate. The output is defined as

$$\mathbf{y}(t) = \mathbf{W}^{out}[1; \mathbf{u}(t); \mathbf{x}(t)] \tag{8}$$

where $\mathbf{y}(t) \in \mathbb{R}^{N_y}$ is network output and $\mathbf{W}^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$ is the output weight matrix.

Lukoševičius [6] writes that the reservoir serves two functions:

1. as a high-dimensional nonlinear expansion, similarly to kernel methods, where input in input space is not linearly separable and by projecting in a higher dimensional space it may become linearly separable.

2. as a dynamical short-term memory.

2.2.2 Echo state property

Jaeger in [3] defined the term *echo state property*, which is a necessary condition for the echo state network to work. The definition for a network with no output feedback is as follows:

Definition 2 *Assume that input is drawn from a compact input space U and network states lie in a compact set A. Assume that the network has no output feedback connections. Then, the network has echo states, if the network state $\mathbf{x}(n)$ is uniquely determined by any left-infinite input sequence $\bar{\mathbf{u}}^{-\infty}$. More precisely, this means that for every input sequence $\ldots, \mathbf{u}(n-1), \mathbf{u}(n) \in \mathrm{U}^{-\mathbb{N}}$, for all state sequences $\ldots, \mathbf{x}(n-1), \mathbf{x}(n)$ and $\mathbf{x}'(n-1), \mathbf{x}'(n) \in \mathrm{A}^{-\mathbb{N}}$, where $\mathbf{x}(i) = \mathrm{T}(\mathbf{x}(i-1), \mathbf{u}(i))$ and $\mathbf{x}'(i) = \mathrm{T}(\mathbf{x}'(i-1), \mathbf{u}(i))$, it holds that $\mathbf{x}(n) = \mathbf{x}'(n)$.*

2.2.3 Hyperparameters

The specific instance of an echo state network is defined by a set of parameters, namely $(\mathbf{W}^{in}, \mathbf{W}, \alpha)$. In the next paragraphs we will outline the principles for choosing these parameters. We are going to follow the recommendations according to Lukoševičius [6].

1. **Reservoir matrix W**

   - *Size of reservoir* - the consensus is that the larger number of neurons in the reservoir the better, but on the other there is danger of over-parametrization and over-fitting. In general

     $$T >= 1 + N_x + N_u$$

     where T is the size of the dataset, N_x is the number of neurons in reservoir, and N_u is the dimension of the input.

   - *Distribution of reservoir weights* - at the initialisation step of the algorithm the elements of the matrix $\mathbf{W}$ are randomly generated and not much manipulated afterwards. The weights are drawn independently from uniform or normal distribution symmetrical around zero.

   - *Spectral radius* - as was stated in section 2.2.2 the existence of the echo state property in an echo state network is essential for the network to work. Jaeger [3] states the network has no echo state when the spectral radius of the reservoir matrix is $|\lambda_{max}(\mathbf{W})| > 1$, the admissible state set is $[-1, 1]^N$ and if the input set contains 0. Thus is standard practice to scale the reservoir matrix $\mathbf{W}$ so that $|\lambda_{max}(\mathbf{W})| < 1$. The specific value $|\lambda_{max}(\mathbf{W})|$ has to determined experimentally in way that maximizes performance.

2. **Input matrix $\mathbf{W}^{in}$**

   Similarly to the initialization of the reservoir matrix $\mathbf{W}$, the elements of input matrix $\mathbf{W}^{in}$ randomly selected either from uniform or normal symmetrical distribution.

3. **Leaking rate α**

Assume a dynamical system

$$\dot{\mathbf{x}} = -x + \tanh(\mathbf{W}^{in}[1; \mathbf{u}] + \mathbf{W}\mathbf{x})$$

and using a discretization of the system

$$\dot{\mathbf{x}} \approx \frac{\mathbf{x}(t) - \mathbf{x}(t-1)}{\triangle t}$$

we get

$$\mathbf{x}(t) = (1 - \triangle t)\mathbf{x}(t-1) + \triangle t \tanh(\mathbf{W}^{in}[1; \mathbf{u}(t)] + \mathbf{W}\mathbf{x}(t-1)). \tag{9}$$

By comparing equations (7) and (9) we can regard the leaking rate parameter as the size of the discrete time step and scale accordingly. In practice this parameter has to be determined experimentally.

2.2.4 Learning process

As can be seen from previous sections there hasn't been any training involved, yet. The the only training is in in finding the output layer weight matrix $\mathbf{W}^{out}$. Precisely solving

$$\mathbf{Y}^{target} = \mathbf{W}^{out}\mathbf{X} \tag{10}$$

where $\mathbf{Y}^{target} \in \mathbb{R}^{N_y \times T}$, $\mathbf{X} \in \mathbb{R}^{(1+N_u+N_x)\times T}$ and T the size of the training set. Lukoševičius [6] proposes to use regression with Tikhonov regularization

$$\mathbf{W}^{out} = \mathbf{Y}^{target}\mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}} + \beta\mathbf{I})^{-1} \tag{11}$$

where $\beta \in \mathbb{R}$ is the regularization parameter and has to be determined experimentally. When $\beta = 0$ we get

$$\mathbf{W}^{out} = \mathbf{Y}^{target}\mathbf{X}^{+} \tag{12}$$

where $\mathbf{X}^{+} = \mathbf{X}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})^{-1}$ is the right Moore-Penrose inverse.

2.2.5 Short term memory capacity

As has been already stated, due to the feedback loops, echo state networks can be regarded as a dynamic short-term memory. Jaeger [4] defined a measure to quantify the reservoirs ability to store and recall past input. It is called the short-term memory capacity:

Definition 3 *Let $\nu(n) \in \mathrm{U}$ (where $-\infty < n < +\infty$ and $\mathrm{U} \in \mathbb{R}$ is a compact interval) be a single-channel stationary input signal. Assume that we have a recurrent neural network, specified by its internal weight matrix $\mathbf{W}$, its input weight (column) vector $\mathbf{w}^{in}$ and the unit output functions $\mathbf{f}, \mathbf{f}^{out}$. The network receives $\nu(n)$ at its input unit. For a given delay k and an output unit y_k with connection weight (row) vector $\mathbf{w}_k^{out}$ we consider the determination coefficient*

$$\begin{aligned} d[\mathbf{w}_k^{out}](\nu(n-k), y_k(n)) = & \\ = d\left(\nu(n-k), \mathbf{w}_k^{out}\begin{pmatrix}\nu(n)\\ \mathbf{x}(n)\end{pmatrix}\right) & \\ = \frac{cov^2(\nu(n-k), y_k(n))}{\sigma^2(\nu(n))\sigma^2(y_k(n)))} & \end{aligned} \tag{13}$$

where cov denotes covariance and σ^2 variance.

1. *The k-delay short term memory capacity of the network is defined by*

$$MC_k = \max_{\mathbf{w}_k^{out}} d[\mathbf{w}_k^{out}](\nu(n-k), y_k(n)). \tag{14}$$

2. *The short term memory capacity of the network is*

$$MC = \sum_{k=1}^{\infty} MC_k. \tag{15}$$

Jaeger in [4] also proved a theoretical limit for the short-term memory capacity:

Proposition 1 *The memory capacity for recalling i.i.d input by a N-unit recurrent neural network with linear output units is bounded by N.*

2.2.6 Lyapunov characteristic exponent

The reservoir of an echo state network can be looked at as a discrete dynamical system and the performance of the the whole network depends on its stability. A popular measure of the degree of the instability of a dynamical system is called *the maximum Lyapunov characteristic exponent.*

Definition 4 *Let*

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t)) \tag{16}$$

be a discrete time dynamical system and

$$\mathbf{x}^{'}(0) = \mathbf{x}(0) + \delta\mathbf{x}(0) \tag{17}$$

be the initial conditions a trajectory $\mathbf{x}^{'}(t)$ *obtained by an infinitesimal displacement from* $\mathbf{x}(0)$ *such that* $\gamma_0 = \|\delta\mathbf{x}(0)\| \ll 1$. *Then the maximum Lyapunov characteristic exponent is*

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \ln\left(\frac{\gamma_t}{\gamma_0}\right) < \infty \tag{18}$$

where $\gamma_t = \|\mathbf{x}^{'}(t) - \mathbf{x}(t)\|$.

From the approximation

$$\gamma_t \sim \gamma_0 e^{\lambda t} \tag{19}$$

it it can be seen that when $\lambda > 0$ the system is sensitive to initial conditions and therefore is chaotic. For $\lambda < 0$ the system is sub-critical. The value $\lambda \approx 0$ is when phase transition occurs and is often referred to as the edge of criticality or critical point.

For numerical computation of λ we implement a popular method according to Cencini et.al [7].

1. We start with the infinitesimal perturbation γ_0 and evolve to system one time step ahead, thus getting a normalized tangent vector $\mathbf{w}(1) = \frac{\mathbf{x}^{'}(1)-\mathbf{x}(1)}{\gamma_0}$ and setting $\alpha(1) = \|\mathbf{w}(1)\|$.

2. Next we rescale $\mathbf{w}(1)$ to $\frac{\mathbf{w}(1)}{\|\mathbf{w}(1)\|}$ and evolve system one step ahead again. We repeat this process and store the amplitudes $\alpha(t) = \|\mathbf{w}(t)\|$.

3. The maximum Lyapunov exponent is obtained as:

$$\lambda = \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} \ln(\alpha(i)). \tag{20}$$

In the echo state network setting we set the infinitesimal perturbation to every neuron unit individually and compute λ_i of the $i-th$ perturbed for every $i = 1, \ldots, N$ of the N unit reservoir. The final estimated is computed as average $\lambda =< \lambda_i >$.

2.2.7 Orthogonalization and orthonormalization of reservoir

We have already mentioned reservoir matrix scaling according to the desired spectral radius. Another ways to scale the reservoir matrix is in relation to orthogonality. More precisely to satisfy the condition $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$. Farkaš et. al. [8] proposed a gradient based orthogonalization and orthonormalization methods and have shown that the theoretical limits in the memory capacity task, proved by Jaeger [4], can be achieved via this methods.

The update formula in case of the orthogonalization method is

$$\triangle\, \mathbf{w}_i = -\eta \frac{4}{\|\mathbf{w}_i\|}(\mathbf{I} - \tilde{\mathbf{w}}_i \tilde{\mathbf{w}}_i^\top)(\tilde{\mathbf{W}}\tilde{\mathbf{W}}^\top)\tilde{\mathbf{w}}_i \tag{21}$$

where η is the learning rate and $\tilde{\mathbf{W}}$ denotes matrix $\mathbf{W}$ with normalized columns.

The update formula for the orthonormalization method is

$$\triangle\, \mathbf{w}_i = -\eta\ 4\ (\mathbf{W}\mathbf{W}^\top\mathbf{W} - \mathbf{W}). \tag{22}$$

2.3 Information measures

2.3.1 Introduction to information theory

In order to get an insight into transfer entropy and active information storage we have to define some key information theoretic concepts. We will follow the definitions according to MacKay [9]. At first we define the Shannon information content and entropy of a discrete random variable X.

Definition 5 *Shannon information content of an outcome x is defined to be*

$$h(x) = \log_2 \frac{1}{P(x)}. \tag{23}$$

The units are called bits.

Definition 6 *The entropy of an ensemble X is defined to be the average Shannon information content of an outcome:*

$$\begin{aligned} H(X) &= \sum_{x \in \mathcal{A}_X} P(x) \log \frac{1}{P(x)} \quad \textit{for } P(x) \neq 0 \\ &= 0 \qquad\qquad\qquad\qquad\quad \textit{for } P(x) = 0 \end{aligned} \tag{24}$$

since $\lim_{\theta \to 0^+} \theta \log \frac{1}{\theta} = 0$.

Next we need to define conditional entropy.

Definition 7 *The conditional entropy of X given $y = b_k$ is the entropy of the probability distribution $P(x \mid y = b_k)$.*

$$H(X \mid y = b_k) = \sum_{x \in \mathcal{A}_X} P(x \mid y = b_k) \log \frac{1}{P(x \mid y = b_k)} \tag{25}$$

Definition 8 *The conditional entropy of X given Y is average, over y, of the conditional entropy of X given y.*

$$\begin{aligned} H(X \mid Y) &= \sum_{y \in \mathcal{A}_y} P(y) \left[\sum_{x \in \mathcal{A}_X} P(x \mid y) \log \frac{1}{P(x \mid y)} \right] \\ &= \sum_{x \in \mathcal{A}_X, y \in \mathcal{A}_Y} P(x, y) \log \frac{1}{P(x \mid y)}. \end{aligned} \tag{26}$$

The interpretation of conditional transfer entropy is that it measures the average uncertainty that remains about X when Y is known.

Now we can define mutual information and conditional mutual information.

Definition 9 *The mutual information between X and Y is*

$$I(X; Y) = H(X) - H(X \mid Y). \tag{27}$$

Mutual information measures the reduction of uncertainty about X that results from learning the value of y.

Definition 10 *The conditional mutual information between X and Y given $z = c_k$ is the mutual information between the random variables X and Y in the joint ensemble $P(x, y \mid z = c_k)$,*

$$I(X; Y \mid z = c_k) = H(X \mid z = c_k) - H(X \mid Y, z = c_k). \tag{28}$$

Definition 11 *The conditional mutual information between X and Y given $z = c_k$ is the mutual information between the random variables X and Y in the joint ensemble $P(x, y \mid z = c_k)$,*

$$I(X; Y \mid z = c_k) = H(X \mid z = c_k) - H(X \mid Y, z = c_k). \tag{29}$$

Definition 12 *The conditional mutual information between X and Y given Z is the average over z of the conditional mutual information from Definition 11.*

$$I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z). \tag{30}$$

Entropy and all other measures based on entropy we have just introduced are defined for a discrete set of probabilities. Shannon in [10] analogously defined the entropy of a continuous distribution with the density distribution function $p(x)$ as

$$H(X) = \int_{-\infty}^{\infty} p(x) \log \frac{1}{p(x)} dx. \tag{31}$$

This quantity is sometimes called *differential entropy* although this is not a measure of uncertainty of the random variable X as Shannon intended it to be.

Similarly, from the identity

$$I(X;Y) = \sum_{x \in \mathcal{A}_X, y \in \mathcal{A}_Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \tag{32}$$

we can define mutual information of a continuous variable as:

$$I(X;Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy. \tag{33}$$

2.3.2 Limiting density of discrete points

As it has been already noted in the previous section, Shannon's formulation of entropy for continuous variables is not an information measure. First it lacks many properties of discrete entropy and second it is not a result of any proper derivation. Nevertheless differential entropy has many theoretical applications.

Jaynes in [20] proposed a different approach to defining entropy of a continuous variable. Let $x_i, i = 1, \ldots, n$ be discrete points such that

$$\lim_{n \to \infty} \frac{1}{n} (\text{number of points in a} < x < \text{b}) = \int_a^b m(x) dx \tag{34}$$

exists. Then the differences $(x_{i+1} - x_i)$ in the neighbourhood of any particular value of x will tend to zero so that

$$\lim_{n \to \infty} n(x_{i+1} - x_i) = m(x_i)^{-1}. \tag{35}$$

The discrete probability $P(x_i)$ from Definition 6 will transform to

$$P(x_i) = p(x_i)(x_{i+1} - x_i). \tag{36}$$

Equivalently from equation (35)

$$P(x_i) \to p(x_i)\frac{1}{nm(x_i)}. \tag{37}$$

Plugging (36) and (37) into the definition of Shannon entropy Definition 6 we get

$$H(X) \to -\int p(x)log\left[\frac{p(x)}{nm(x)}\right]dx. \tag{38}$$

Following this reasoning, Jaynes defined the continuous information measure as:

$$\overline{H}(X) = \lim_{n\to\infty}\left[H(X) - \log(n)\right] = -\int p(x)log\left[\frac{p(x)}{m(x)}\right]dx. \tag{39}$$

2.3.3 Transfer entropy and active information storage

Let $\{x(t)\}$ and $\{y(t)\}$ be the realizations of two processes $\{X(t)\}$ and $\{Y(t)\}$. Paluš in [11] proposed, as an approach to measure the directional information rate, to measure conditional mutual information $I(y(t); x(t+\tau) \mid x(t))$. It is the average amount of information contained in the process $\{Y(t)\}$ about the process $\{X(t)\}$ in its future τ time units ahead conditioned on $X(t)$, as opposed to the mutual information $I(y(t); x(t+\tau)$ which can contain information about $X(t+\tau)$ in $X(t)$. Similarly Shreiber in [12] defines mutual information rate of two Markov processes $\{I\}$ and $\{J\}$ of order k and l by measuring the deviation from independence given by the Markov property $p(i_{t+1} \mid i_t^{(k)}) = p(i_{t+1} \mid i_t^{(k)}, j_t^{(l)})$ using conditional Kullback-Leibler divergence in the following form

$$D_{KL}(P(I_{t+1} \mid I_t^{(k)}, J_t^{(l)}) \| P(I_{t+1} \mid I_t^{(k)}, J_t^{(l)})) =$$
$$= \sum p(i_{t+1}, i_t^{(k)}, j_t^{(l)}) \log \frac{p(i_{t+1} \mid i_t^{(k)}, j_t^{(l)})}{p(i_{t+1} \mid i_t^{(k)})}.$$

These ideas inspired the current definition of transfer entropy.

Definition 13 *Let X_t and Y_t be two processes. The transfer entropy from the source Y with the history length k to the target X with history length l is*

$$T_{Y\to X}^{(k,l)} = I(X_t\ ; \mathbf{Y}_{t-1}^{(l)} \mid \mathbf{X}_{t-1}^{(k)}) \tag{40}$$

where

$$\mathbf{X}_{t-1}^{(k)} = (X_{t-1}, X_{t-1-\tau_k}, X_{t-1-2\tau_k}, ..., , X_{t-1-(k-1)\tau_k})$$
$$\mathbf{Y}_{t-1}^{(l)} = (Y_{t-1}, Y_{t-1-\tau_l}, Y_{t-1-2\tau_l}, ..., , Y_{t-1-(l-1)\tau_l}).$$

From the identity for conditional mutual information

$$I(X;Y \mid Z) = H(X) - I(X;Z) - H(X \mid Y, Z)$$

it follow that transfer entropy can be expressed as

$$T^{(k,l)}_{Y\to X} = H(X_t) - I(X_t \mid \mathbf{X}^{(k)}_{t-1}) + H(X_t \mid \mathbf{Y}^{(l)}_{t-1}, \mathbf{X}^{(k)}_{t-1}).$$

The second term on the right in the expression above has a special significance for us.

Definition 14 *The active information storage of the process X with history length k is*

$$A^{(k)}_X = I(\mathbf{X}^{(k)}_{t-1}; X_t) = H(X_t) - H(X_t \mid \mathbf{X}^{(k)}_{t-1}). \tag{41}$$

The active information storage measures the amount of information in the past state of $\mathbf{X}^{(k)}_t$ of X about its next value X_t.

2.3.4 Transfer entropy and Granger causality

Transfer entropy is closely related to another concept of dependency, namely Granger causality. This relationship provides a different interpretation of the concept of transfer entropy. Lets look take a closer look at it.

We take the definition of Granger causality from [13]

Definition 15 *Let us use the notation from Definition 13. Let $F(x_t|\mathbf{x}^{(k)}_{t-1}, \mathbf{y}^{(l)}_{t-1})$ be the distribution function of the target variable X_t conditional on $\mathbf{X}^{(k)}_{t-1}, \mathbf{Y}^{(l)}_{t-1}$ and $F(x_t|\mathbf{x}^{(k)}_{t-1})$ be the distribution function of X_t conditional on its own past, then variable Y is said to Granger-cause variable X if*

$$F(x_t|\mathbf{x}^{(k)}_{t-1}, \mathbf{y}^{(l)}_{y-1}) \neq F(x_t|\mathbf{x}^{(k)}_{t-1}). \tag{42}$$

Now, consider tow linear regression models

$$X_t = X_{t-1}A_1 + \ldots + X_{t-k}A_k + Y_{t-1}B_1 + \ldots + Y_{t-l}B_l + \epsilon_t \tag{43}$$

$$X_t = X_{t-1}A^{'}_1 + \ldots + X_{t-k}A^{'}_k + \epsilon^{'}_t \tag{44}$$

with parameters of the models $A_i, A^{'}_i, B^{'}_j$. Geweke in [18] defined the measure of Granger causality from Y to X the following way,

$$F^{(k,l)}_{Y\to X} = \log(|\mathrm{var}(\epsilon^{'}_t)|/|\mathrm{var}(\epsilon_t)|) \tag{45}$$

where $|\cdot|$ denotes the determinant. We can observe that this is actually log-likelihood ratio test under the null hypothesis

$$H_0 : B_1 = \cdots = B_l = 0 \tag{46}$$

when the residuals of the both model is gaussian.

From what has already been written it can be seen that both transfer entropy and Granger causality are measures of predictive causality. The similarity is even closer when we consider the joint process X_t, Y_t as multivariate gaussian. Barnett et. al. in [19] proved that under these conditions that Granger causality and transfer entropy are equivalent up to factor 2 i.e.

$$F^{(k,l)}_{Y\to X} = 2T^{(k,l)}_{Y\to X}. \tag{47}$$

2.3.5 Estimation of transfer entropy and active information storage

Estimation of entropy measures is an open problem. Currently there are few available classes of estimators that one can choose from, based on the properties of the observed data. Transfer entropy is no exception and the best estimator given some specific criteria is yet to be determined. An overview of transfer entropy estimators can be found in [13] or [14].

We are going to focus on one specific estimator of the class of estimators based on k-nearest neighbour search.

Kozachenko-Leonenko Shannon entropy estimator We are going to put the basic idea of behind Kozachenko-Leonenko differential entropy estimator as was presented in [15].

Let X be a continuous random variable, $f(x)$ be its density and its differential entropy as defined in (9). Specifically

$$H(X) = \int_{-\infty}^{\infty} f(x) \log \frac{1}{f(x)} dx. \tag{48}$$

Then the Monte-Carlo estimate of $H(X)$ is

$$\widehat{H}(X) = \frac{1}{N} \sum_{i=1}^{N} \log \frac{1}{f(x_i)}. \tag{49}$$

Since we don't know $f(x_i)$ it has to be substituted by an estimate $\widehat{f}(x_i)$ which we are going to find using k-nearest neighbours of x_i.

Let $P_k(\epsilon)$ be the probability distribution of the distance between x_i and its kth nearest neighbour. Then $P_k(\epsilon)d\epsilon$ is the probability that there is one point in the r distance from x_i where $r \in < \frac{\epsilon}{2}; \frac{\epsilon}{2} + \frac{d\epsilon}{2} >$, $k-1$ points are at distances less than r and $N-k-1$ points are at distances greater than kth nearest neighbour. Using multinomial distribution formula we get

$$P_k(\epsilon)d\epsilon = \frac{(N-1)!}{1!(k-1!)(N-k-1)!} \left(\frac{dp_i(\epsilon)}{d\epsilon} d\epsilon \right) (p_i(\epsilon))^{k-1} (1-p_i(\epsilon))^{N-k-1} \tag{50}$$

where $p_i(\epsilon)$ is the probability mass of ϵ ball centered at x_i, that is

$$p_i(\epsilon) = \int_{\|\xi - x_i\| < \frac{\epsilon}{2}} f(\xi) d\xi. \tag{51}$$

It follows from (50) and (51) that the expectation value $\log p_i(\epsilon)$ is given by

$$\begin{aligned} E(\log p_i(\epsilon)) &= \int_0^\infty \log p_i(\epsilon) P_k(\epsilon) d\epsilon \\ &= \psi(k) - \psi(N) \end{aligned} \tag{52}$$

where $\psi(x)$ is the digamma function.

If we assume that $f(x)$ is constant in the entire ϵ ball we can approximate $p_i(\epsilon)$ by

$$p_i(\epsilon) \approx c_d \epsilon^d f(x_i), \tag{53}$$

where d is the dimension of x and c_d is the volume of the d-dimensional unit ball. For the maximum norm $c_d = 1$.

Finally taking the logarithm and expectation of (53), and combining it with (52) and (49) we get Kozachenko-Leonenko entropy estimator

$$\widehat{H}(X) = -\psi(k) + \psi(N) + \log c_d + \frac{d}{N} \sum_{i=1}^{N} \log \epsilon(i), \tag{54}$$

where $\epsilon(i)$ is twice the distance from x_i to its kth nearest neighbour.

Kraskov-Stögbauer-Grassberger mutual information estimator Kraskov, Stögbauer and Grassberger in [15] came with a method of using Kozachenko-Leoneko entropy estimator to estimate mutual information. Using the identity

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{55}$$

we are going to obtain an estimator $\widehat{I}(X,Y)$.

First we directly apply Kozachenko-Leonenko entropy estimator to joint random variable $Z=(X,Y)$ with maximum norm and we get

$$\widehat{H}(X,Y) = -\psi(k) + \psi(N) + \frac{d_X + d_Y}{N}\sum_{i=1}^{N} \log \epsilon(i) \tag{56}$$

where $\epsilon(i)$ is the $\frac{\epsilon}{2}$ from z_i to its kth nearest neighbour and $d_Z = d_X + d_Y$.

For the estimate of $H(X)$ we take the distance $\epsilon(i)$ from (56) as an approximation of $[n_x(i)+1]$st nearest neighbour of x_i, where $n_x(i)$ is the number of points in within $\|x_j - x_i\| < \frac{\epsilon}{2}$, and we get

$$\widehat{H}(X) = -\frac{1}{N}\sum_{i=1}^{N} \psi(n_x(i)+1) + \psi(N) + \frac{d_X}{N}\sum \log \epsilon(i). \tag{57}$$

Analogously for the marginal space Y we get

$$\widehat{H}(Y) = -\frac{1}{N}\sum_{i=1}^{N} \psi(n_y(i)+1) + \psi(N) + \frac{d_Y}{N}\sum_{i=1}^{N} \log \epsilon(i). \tag{58}$$

Combining (55), (56), (57) and (58) we get the Kraskov, Stögbauer and Grassberger estimator

$$I^{(1)}(X,Y) = \psi(k) - \langle \psi(n_x+1) + \psi(n_y+1)\rangle + \psi(N) \tag{59}$$

where $\langle \cdots \rangle = \frac{1}{N}\sum_{i=1}^{N}(\cdots)$.

Estimating transfer entropy and active information storage estimator Since active information storage of a process is defined in Definition 14 as mutual information between its current and past state, it follows from (59) that the active information estimator is written as

$$\widehat{A}_X^{(k)} = \psi(k) + \psi(N) - \left\langle \psi(n_{\mathbf{x}_{t-1}^{(k)}} + 1) + \psi(n_{x_t}+1)\right\rangle. \tag{60}$$

As for an estimate of transfer entropy Gomez-Herrero et. al. [16] generalized the idea of Kraskov et. al. [15]. Let $V=(V_1,\dots,V_m)$ be a random m-dimensional vector. Then the entropy combination is defined as

$$C(V_{\mathcal{L}_1},\dots,V_{\mathcal{L}_p}) = \sum_{i=1}^{p} s_i H(V_{\mathcal{L}_1}) - H(V) \tag{61}$$

where $\forall i \in < i, p >$: $\mathcal{L}_i \subset < 1, m >$ and $s_i \in \{-1, 1\}$ such that $\sum_{i=1}^{p} s_i \chi_{\mathcal{L}_i} = \chi_{<1,m>}$ where χ_S is characteristic function of S and the entropy combination estimator is given by

$$\widehat{C}(V_{\mathcal{L}_1}, \dots, V_{\mathcal{L}_p}) = F(k) - \sum_{i=1}^{p} s_i \left\langle F(k_i(j)) \right\rangle \tag{62}$$

where $F(k) = \psi(k) - \psi(N)$ and $k_i(j) = n_{V_{\mathcal{L}_i}}(j) + 1$ for $j = 1, \dots, N$ as was formulated in section 2.3.5.

For example, from equation (55) it can be seen that mutual information is an entropy combination. Similarly from the one of the expressions for conditional mutual information

$$I(X; Y \mid Z) = H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z) \tag{63}$$

it follows that $I(X; Y \mid Z)$ is also an entropy combination and combining (62) and (63) we get the Kraskov, Stögbauer and Grassberger estimator for conditional mutual information

$$\widehat{I}(X; Y \mid Z) = \psi(k) - \left\langle \psi(n_{xz} + 1) + \psi(n_{yz} + 1) - \psi(n_z + 1) \right\rangle . \tag{64}$$

Finally directly from the definition 13 we get the Kraskov, Stögbauer and Grassberger transfer entropy estimator

$$\widehat{T}^{(k,l)}_{Y \to X} = \psi(k) - \left\langle \psi(n_{x_t, \mathbf{x}^k_{t-1}} + 1) + \psi(n_{\mathbf{y}^l_{t-1}, \mathbf{x}^k_{t-1}} + 1) - \psi(n_{\mathbf{x}^k_{t-1}} + 1) \right\rangle . \tag{65}$$

2.3.6 Estimators parameter determination

The Kraskov, Stögbauer and Grassberger belongs to a class of nonparametric estimation methods. Nevertheless, since its based on k-nearest neighbour searches, k in the nearest neighbour algorithm is one of the parameters we have to chose. This parameter has to be empirically determined. Kraskov et. al. [15] suggest $k > 1$ but not too large because of the increase of systematic errors. Generally they propose to use $k = 2 - 4$. Bossomaier et. al. in [13] writes that for $k \geq 4$ the estimator is robust and Wibral in [14] writes that $k = 4$ has been determined as a good choice for ECoG data.

Another problem is determining the embedding vector as was defined in Definition 13. For example, in case of transfer entropy we would like as much of the information that is contained in the target process to condition out. If we won't do this, the measured information transfer might be due to self prediction of the target process and not the dependance of the source and target processes. The method for finding the optimal values for the embedding vector is also called state space reconstruction.

There are multiple methods to determine the state space vectors. One way is to set (m, τ) such that it maximizes the active information storage. Wibral et. al. [14] propose to optimize the embedding parameters using the Ragwitz-Kantz criterion.

Ragwitz and Kantz in [17] proposed a method to extract a Markov process of order $m > 1$ from observed scalar time series using locally constant predictors. In the following paragraphs we will briefly describe this method.

Let $\vec{s}_n = (s_n, s_{n-\tau}, \dots, s_{n-(m-1)\tau})$ be time delay embedding vector such that $s_{n+1} = g(\vec{s}_n)$ exists. Then the locally constant predictor for the unobserved s_{n+1} is

$$\widehat{s_{n+1}} = \frac{1}{|\mathcal{U}_n|} \sum_{\vec{s}_k \in \mathcal{U}_n} s_{k+1} \tag{66}$$

where $\mathcal{U}_n = \{\vec{s}_k : \|\vec{s}_k - \vec{s}_n\| \leq \epsilon\}$. In other words, the mean of the immediate futures of the ϵ-neighbours of $\vec{s}_n$. Ragwitz and Kantz in [17] argue that locally constant predictor is a predictor based on Markov transition probability assumption $p(s_{n+1}|s_n, s_{n-\tau}, \dots, s_\tau) = p(s_{n+1}|s_n, s_{n-\tau}, \dots, s_{n-(m-1)\tau})$. To get the transition the probabilities $p(s_{n+1}|\vec{s}_n)$, a locally constant approximation is used in the form of $p(s_{k+1}|\vec{s}_k) \approx \widehat{p}(s_{n+1}|\vec{s}_n)\ \forall\ \vec{s}_k \in \mathcal{U}_n$. The justification of locally constant predictor follows then from the idea that if we use the mean square error of predictions $e^2 = \sum_{i=1}^{N}(s_{i+1} - \widehat{s}_{i+1})^2$ then the best estimator of s_{n+1} is

$$\widehat{s}_{n+1} = \int_{\vec{s}_k \in \mathcal{U}_n} s_{k+1} p(s_{k+1}|\vec{s}_n) ds_{k+1} \tag{67}$$

Thus we take those time delay embedding vector parameters as optimal which minimize the locally constant prediction error i.e.

$$(m, \tau) = \underset{m \in \mathbb{Z}, \tau \in \mathbb{Z}}{\arg\min} \left\| \vec{s} - \widehat{\vec{s}} \right\|. \tag{68}$$

2.3.7 Transfer entropy significance testing

One of the problems that plague all transfer entropy estimators is bias. Either systematic errors or statistical errors are both properties of estimators that has to be dealt with. If we observe non-zero value of transfer entropy it can be due to bias or variance and the transfer entropy estimator (65) is no exception. Since Kraskov, Stögbauer, Grassberger method is non-parametric one of the suggested statistical testing options is to use a permutation test [14] [13].

We are going to test the null hypothesis that there is no relationship between the source and target variables. First we have to come with surrogate source variable under to assumption that the null hypothesis is true. One way to do it is to by permuting $\mathbf{y}_{t-1}^{(l)}$ in the joint probability space $\{x_t, \mathbf{x}_{t-1}^{(k)}, \mathbf{y}_{t-1}^{(l)}\}$ thus destroying any predictive dependence of Y to X, and computing $T_{Y_{sur} \to X}$ of the surrogate source variable Y_{sur}. Repeating this procedure multiple times we can compute one-sided p-value of the test by the following equation:

$$P(T_{Y_{sur} \to X} \geq T_{Y \to X}) = \sum_{Y_{sur} : T_{Y_{sur} \to X} \geq T_{Y \to X}} P(Y_{sur}). \tag{69}$$

If the p-value is less 0.05 we reject the null hypothesis at the 5% significance level.

3 Experiments

Our aim was to explore the information flow within the ESN reservoir and its relationship to specified task performance. For this reason a specific ESN model had to be chosen. This meant fixing the hyperparameters of the model. At first we wanted to extend the work done by Boedecker et. al. [1].

3.1 Experimental setup

For comparability reasons and consistence of results we choose the same ESN model and also two benchmark tasks used by Boedecker et. al. [1]. The benchmark tasks we used is memory capacity task, and one step ahead prediction of one stochastic and two deterministic processes.

3.1.1 ESN setup

We used ESNs with $N = 100$ reservoir units. A single input neuron and Q output neurons depending on the task. 120 output neurons for the memory capacity task and 1 output neuron for the prediction tasks. We didn't use a bias member in the input and output layers, as well as no direct input–output connections. The leaking rate in (7) was set to $\alpha = 0$. The input weight matrix was initialized from uniform distribution with parameters $\mathcal{U}(-0.1; 0.1)$. The elements of the reservoir matrix $\mathbf{W}$ were drawn from normal distribution with parameters $\mathcal{N}(0; 0.5)$. Concerning the learning process, we set $\beta = 0$ in the Tikhonov regularization in expression (11), effectively getting direct pseudoinverse computation of the readout matrix $\mathbf{W^{out}}$. We discarded the first 100 samples of the reservoir neuron activations in order to get the network running and to get rid of the transients, and 1000 samples of the reservoir neuron activations along with desired outputs were used to set the readout weights. Next 2000 samples from the network output were used to test the networks performance.

3.1.2 Benchmark tasks

In our experiments we used four standard benchmark tasks that are used by the reservoir computing community.

1. **Memory capacity (MC)**

In order to measure the networks ability to recall past input, a sequence of independent identically distributed real numbers drawn from uniform distribution on the closed interval $[-1;1]$ was used as the driving signal.

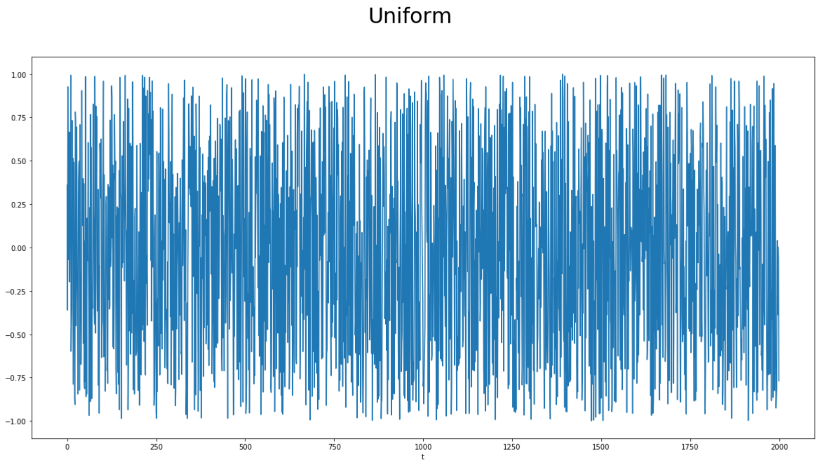

Figure 1: Input used in memory capacity task testing

2. NARMA

We used 30-th order NARMA time series for one time step ahead prediction. The NARMA model was given by the following formula:

$$u(t+1) = 0.2\,u(t) + 0.004\,u(t)\sum_{i=0}^{29} u(t-i) + 1.5\,q(t-29)q(t) + 0.001 \tag{70}$$

were $\forall\ t : q(t) \sim \mathcal{U}(0;0.5)$.

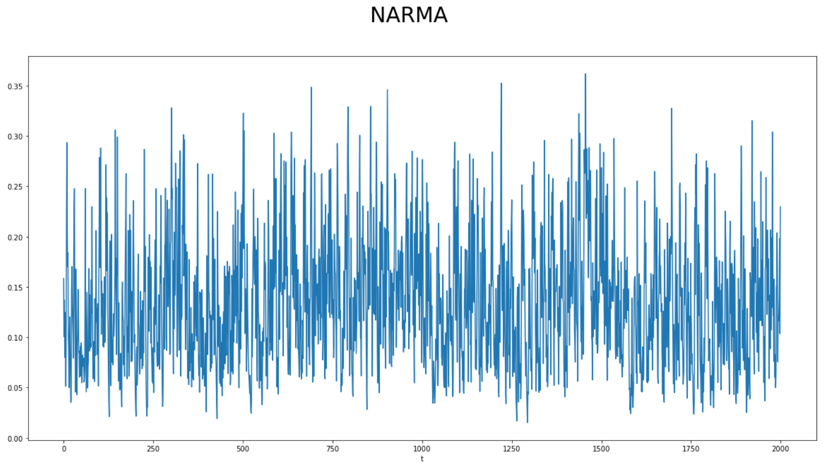

Figure 2: Input used in NARMA prediction task testing.

3. Mackey-Glass system (M–G)

Another standard benchmark task we used is the one time step ahead prediction of the Mackey-Glass system. The system is given by time–delay differential equation

$$\frac{du}{dt} = 0.2\frac{u_{17}}{1+(u_{17})^{10}} - 0.1u \tag{71}$$

were u_{17} is the value of u at time $t-17$.

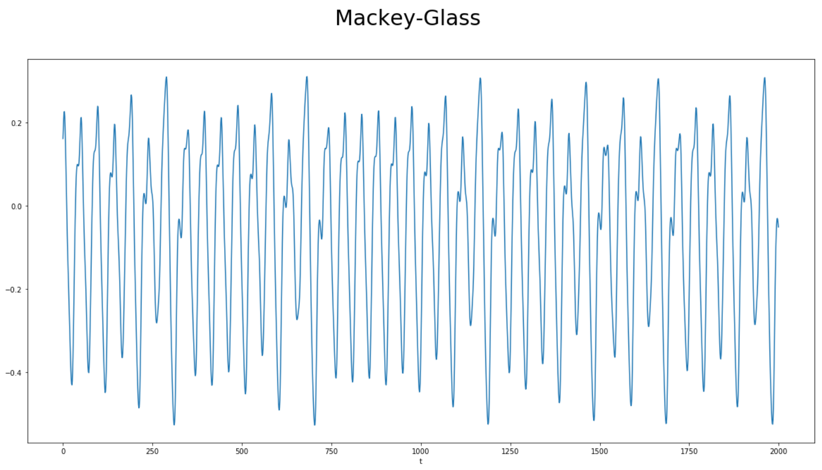

Figure 3: Input used in M–G prediction task testing.

4. **Lorenz system**

   The last task was the one step ahead prediction of the x-coordinate of the Lorenz system given by the following ordinary differential equations:

$$\frac{dx}{dt} = 10(y-x) \tag{72}$$

$$\frac{dy}{dt} = x(28-z) - y \tag{73}$$

$$\frac{dz}{dt} = xy - \frac{8}{3}z. \tag{74}$$

Regarding the assessment of performance, for measuring the memory capacity task performance we used measures introduced in section 2.2.5 and in the cases of prediction tasks we used the normalized root mean square error computed as:

$$\text{NRMSE} = \sqrt{\frac{\langle(\hat{y}(t)-y(t))^2\ \rangle_t}{\langle(y(t)-\langle y(t)\rangle_t)^2\ \rangle_t}} \tag{75}$$

where $\hat{y}(t)$ denotes the predicted value of the process, $y(t)$ is the true value of the process and $\langle\cdot\rangle_t$ denotes the time-average.

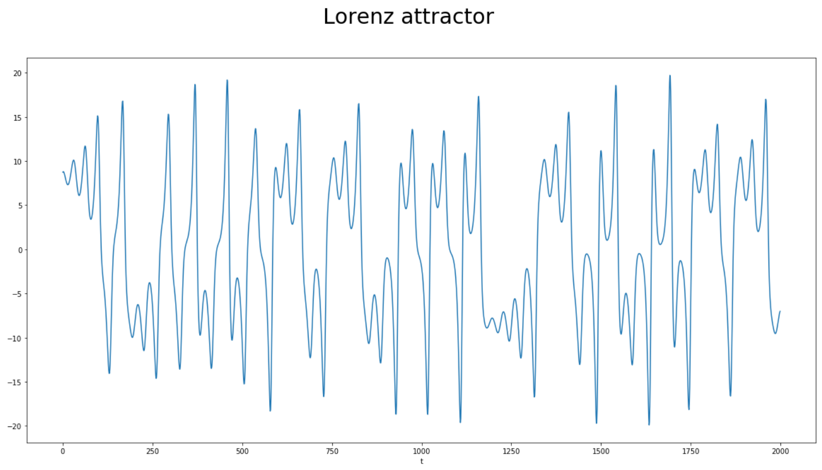

Figure 4: Input used in Lorenz prediction task testing.

3.2 Information measures, performance and stability

In order to explore the relationship between transfer entropy, active information storage and task performance we followed the idea of Boedecker [1] and computed average TE and AIS within the ESN reservoir for different stability modes. To assess the stability of reservoir we computed the maximum Lyapunov exponent introduced in section 2.2.6. Since the maximum Lyapunov exponent can be manipulated only indirectly, we varied the variance σ of the distribution from which the elements of the reservoir matrix were drawn. To get a vide enough spectrum of stability instances we increased σ in such a way that $\log \sigma \in [-1.5; -0.25]$ in 60 steps. For every value of σ (step) we created 5 instances of reservoir matrices. Altogether 130 instances of reservoir matrices and for every such matrix we measured λ, its performance (MC/NRMSE), TE and AIS.

Concerning the setting of the TE and AIS estimators parameters, ideally we would determine them using Ragwitz-Kantz criterion introduced in section 2.3.6 but estimating optimal signal embedding for every neuron for every instance would be computationally demanding. Thus we set the target signal embedding in the TE computation and the signal embedding in AIS computation to $k = 2$ and $\tau_k = 1$, the source signal embedding in the TE computation to $l = 1$ and $\tau_l = 1$. We didn't change these settings in the course of simulations. In the same manner we set the number of nearest neighbour in the k-NN search to 4.

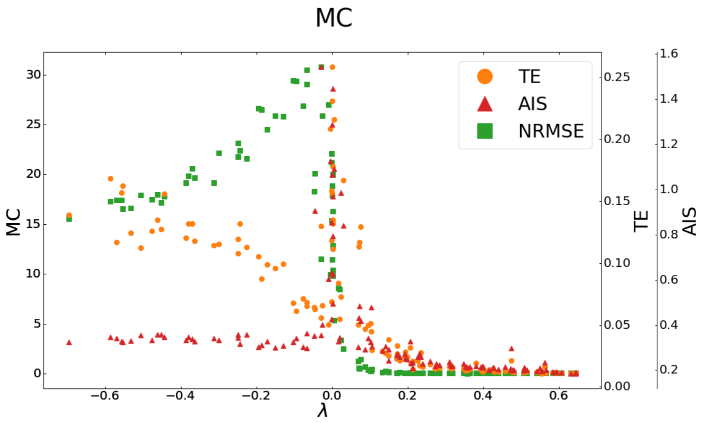

Figure 5: Average values of TE and AIS within the reservoir, and memory capacity in relation to the maximum Lyapunov exponent λ in the MC task. Every datum represents one instance of the reservoir matrix.

3.2.1 MC task

The results in case of MC task are presented in Figure 5. The performance (MC) in this task peaks just before the phase transition from the stable regime to unstable regime. Similarly the information measures peak around the critical point $\lambda \approx 0$. MC, TE and AIS sharply drop to minimal values in the unstable mode ($\lambda > 0$). With decreasing λ in negative range, MC raises and unexpectedly TE falls steadily. AIS does not seem to change accordingly to the changes in TE and MC.

3.2.2 NARMA task

In case of the NARMA task the results are shown in Figure 6. The behavior of information measures and performance (NRMSE) is similar to the MC task. The performance peaks close to the phase transition ($\lambda \approx 0$), as well as TE and AIS, and falls sharply after the phase transition into the the unstable regime. Similarly to the MC task, the performance raises with increasing λ (NRMSE decreases) and also TE in the negative range. AIS doesn't seem to change accordingly.

3.2.3 M–G task

The behavior of information measures and performance in case of M–G task, shown in Figure 7, is quite different from previous tasks. First, the performance does not peak close to the phase transition, but rather in the more stable region ($\lambda \ll 0$) and second TE doesn't seem to change in relation to the performance as in the MC and NARMA

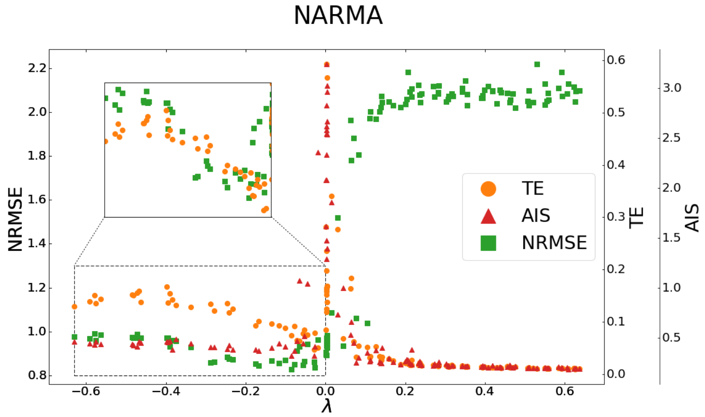

Figure 6: Average values of TE and AIS within the reservoir, and NRMSE in relation to the maximum Lyapunov exponent λ in the NARMA prediction task. Every datum represents one instance of the reservoir matrix.

tasks. On the other hand information measures peak at criticality as in previous cases and fall sharply in the unstable region but AIS stays high for $\lambda > 0$.

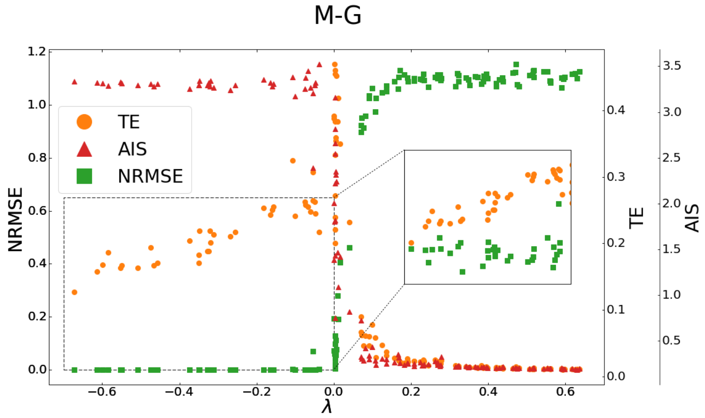

Figure 7: Average values of TE and AIS within the reservoir, and NRMSE in relation to the maximum Lyapunov exponent λ in the M–G prediction task. Every datum represents one instance of the reservoir matrix.

3.2.4 Lorenz task

The results from Lorenz task exploration (Figure 8) resemble the results from the M–G task in that the performance peaks in stable stable regime far from the critical point. TE and AIS also peak further from the critical point than in previous cases. The decrease of performance and information measures from the stable to unstable regime is more gradual than in previous cases indicating that our ESN model is less sensitive to the stability of the reservoir in case of Lorenz driving signal. There seems to be a relationship between TE and performance similar to the MC and NARMA in the stable mode.

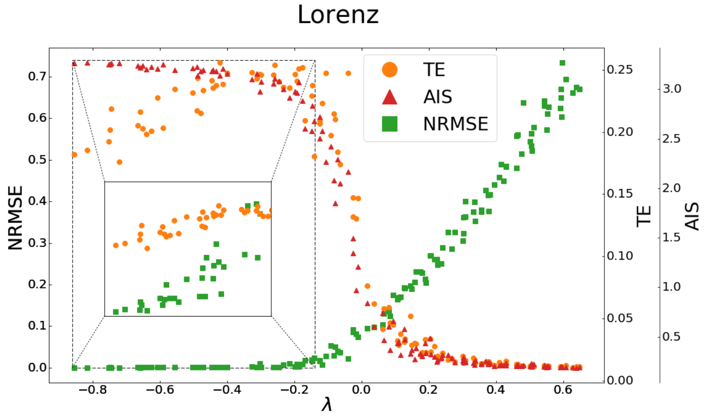

Figure 8: Average values of TE and AIS within the reservoir, and NRMSE in relation to the maximum Lyapunov exponent λ in the Lorenz prediction task. Every datum represents one instance of the reservoir matrix.

3.2.5 Conclusion

From the observed results it seems there is an unexpected relationship between the information measures and performance for λ in stable region. To look at this relationship we computed various correlation coefficients between corresponding information measure and performance. As the transition from the stable to the unstable mode is gradual and task dependent, we looked at the instances for which λ is in stable mode, and such that it maximazes the Pearson correlation coefficient. Thus we gradually decreased the size of the interval beginning with $\lambda = 0$ and for each interval computed correlation coefficients. The results for maximal obtained Pearson correlation coefficients are presented in Table 1 for TE and Table 2 for AIS.

We want to emphasize that the use of the correlation coefficents in such a way is not correct. First the data we compute the correlations do not satisfy the assumptions necessary for statistical testing, and second the idea of looking for maximal correlations by search is questionable. Nevertheless we think it is a good way to explore for potential relationships within the stable region.

In case of TE, our explorations imply that there seems to be negative relationship between information flow and task performance at least for MC, NARMA and Lorenz tasks. That means the higher the performance the lower the overall information flow within the reservoir layer.

Table 1: Values of correlation coefficients between TE and performance (MC or NRMSE depending on task) in stable regimes.

	MC: $\lambda \in$ [-0.6,-0.0064)		NARMA: $\lambda \in$ [-0.6,-0.0017)		M-G: $\lambda \in$ [-0.6,-0.0013)		Lorenz: $\lambda \in$ [-0.9,-0.0233)	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Pearson	-0.920	1.9e-11	0.657	1.0e-05	0.706	9.3e-07	0.664	4.6e-05
Spearman	-0.897	6.6e-13	0.657	1.0e-05	0.240	0.17	0.779	2.4e-07
Kendall	-0.743	8.3e-09	0.444	1.0e-04	0.159	0.17	0.587	3.5e-06

In case of AIS, the selfpredictability of the units in the reservoir seem to depend on the task. For MC task there seems to be a negative relationship between AIS and performance, no relationship in NARMA and M–G tasks, and positive dependence in Lorenz task (note that higher performance means lower NRMSE in NARMA, M–G and Lorenz tasks, and higher MC in MC task). As though the network benefits more from the selfpredictability of each neuron than information exchange between different units in the Lorenz task.

Table 2: Values of correlation coefficients between AIS and performance (MC or NRMSE depending on task) in stable regimes.

	MC: $\lambda \in$ [-0.6,-0.0064)		NARMA: $\lambda \in$ [-0.6,-0.0017)		M-G: $\lambda \in$ [-0.6,-0.0013)		Lorenz: $\lambda \in$ [-0.9,-0.0233)	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Pearson correlation	-0.737	3.4e-06	0.01	0.954	-0.920	2.8e-16	-0.911	1.2e-12
Spearman correlation	-0.708	1.2e-05	0.452	0.005	0.051	0.763	-0.866	3.2e-10
Kendall correlation	-0.513	6.9e-05	0.3	0.009	0.038	0.734	-0.690	4.9e-08

Altogether the results indicate a surprising inverse relationship between TE and performance. To get a more precise answer to this hypothesis its necessary to design the experiment more thoroughly.

3.3 Reservoir neuron signal embedding

In the next step we wanted to get a closer look on information measures at different stability settings but first the values of KSG estimators parameters had to be chosen. As was already mentioned in section 3.2, ideally we would determine the embedding vector and time delay for every single neuron signal and use it in the TE and AIS estimation. This approach would computationally very demanding and the whole reservoir information measure estimation would be overparameterized. Thus we simplified the this task and set the TE and AIS parameters to one value for every neuron in single reservoir.

To find this value we used the Ragwitz-Kantz criterion (section 2.3.6) to determine the embedding vectors for every neuron in 100 instances of reservoir matrices for every task and every scaling. Then the most frequent parameter observation in a speciffic task and scaling was taken as the optimal value in subsequent TE and AIS computations. The search space in the Ragwitz-Kantz method was $k = 1, \ldots, 6$ for the the embedding vector length and $\tau = 1, \ldots, 6$ for the time delay.

The ESNs were initialized according to section 3.1.1. and subsequently the reservoir matrix was scaled in order to get a speciffic spectral radius (section 2.2.3) according to desired stability properties i.e. *unscaled* for unstable, $\rho = 0.6$ for stable and $\rho = 0.95$ close to criticality. We also added two more scalings of the reservoir matrix according to orthogonality/orthonormality introduced in section 2.2.7. We used the spectral radius scaling $\rho = 0.95$ and then ran 30 iterations of the OG method (equation (21)) with the learning rate $\eta = 0.03$. We did the same procedure with the ON method (equation (22)), except that the learning rate was set to $\eta = 0.07 * (0.9)^j$, $j = 0, \ldots, 30$. Altogether we got 5 different scalings.

3.3.1 MC task

The results for the MC task are presented Figure 9. The embedding length and delay exploration have single maxima for all scalings. The maxima seem unsensitive to scalings of the reservoir matrix. Also the most frequent choices of embedding vectors are for the largest length allowed in the exploration, except for the scaling $\rho = 0.6$. If we consider the driving signal in the MC task then it is possible that the single neuron signals can not be approximated by a Markov process and thus $k \to \infty$. Or the maximal allowed length in the exploration was to low.

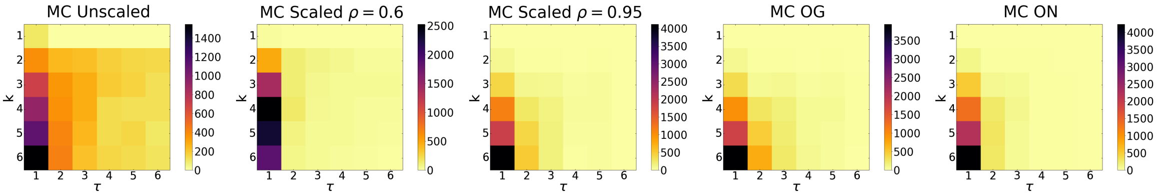

Figure 9: (k, τ) pairs for various scalings of the reservoir in MC task. Each cell denotes the number of occurrences when the locally constant predictor with the setting (k, τ) minimized the prediction error in a single neuron signal.

3.3.2 NARMA task

In the case of NARMA task the results, shown in Figure 10, are almost identical to the result of the MC task. The maxima for every scaling are the same as in the case of MC task. These similarities can be regarded as an indication that the neuron activation signals in the MC and NARMA tasks have some identical properties. For example the order of the underlying Markov process.

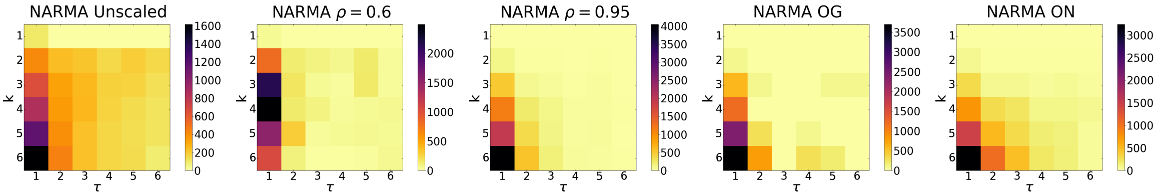

Figure 10: (k, τ) pairs for various scalings of the reservoir in NARMA task. Each cell denotes the number of occurrences when the locally constant predictor with the setting (k, τ) minimized the prediction error in the single neuron signal.

3.3.3 M–G task

The results of the M–G task exploration are presented in Figure 11. In this case there are also single maxima but for the scaled instances there is a possibility of a second node for τ larger then 6. The scalings of the reservoir matrix have a noticeable influence on the embedding vector choice, very different from the MC and NARMA tasks. This also implies that the scaled neuron signals for this task could be approximated by a Markov process of order 2.

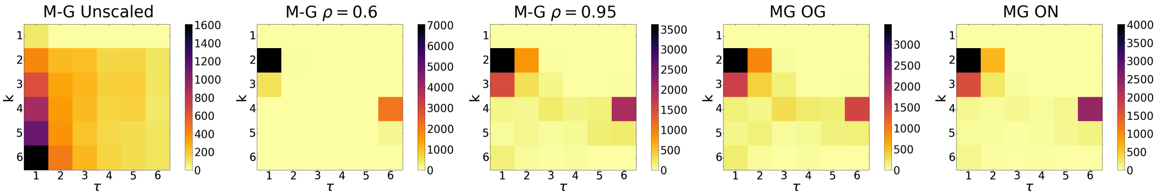

Figure 11: (k, τ) pairs for various scalings of the reservoir in M–G task. Each cell denotes the number of occurrences when the locally constant predictor with the setting (k, τ) minimized the prediction error in the single neuron signal.

3.3.4 Lorenz task

The results for the Lorenz task (Figure 12) are similar to the M–G task in that they have singular maxima nodes and the length of the most frequent embedding vector is 2. The difference is in delay τ and that the distribution of winning (k, τ) pairs is more spread than in previous cases. The higher delay τ could be explained by a small $\triangle t$ time step when the training and validation dataset was created.

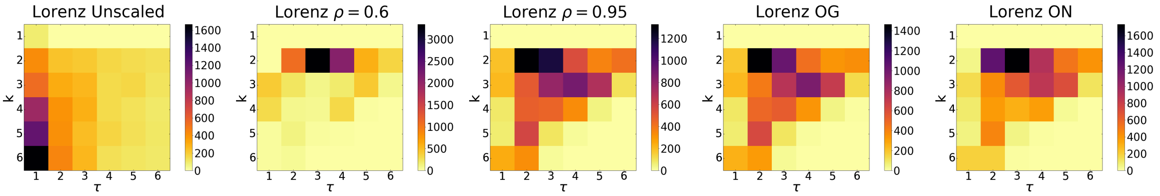

Figure 12: (k, τ) pairs for various scalings of the reservoir in Lorenz task. Each cell denotes the number of occurrences when the locally constant predictor with the setting (k, τ) minimized the prediction error in the single neuron signal.

3.3.5 Conclusion

As a side product of the search for optimal values of the KSG estimators parameters which are presented in Table 3, we did get an insight into the complexity of reservoir neuron activation signals. If we consider the order of an underlying Markov process as a measure of how complex an observed time series is then the reservoir activations in the MC and NARMA tasks have the same complexity, and the same goes for M–G and Lorenz tasks. This was expected since the former are driven by a stochastic signal and the later by a deterministic signal.

Table 3: Most frequent (k, τ) pairs for each task and scaling.

Task	MC		NARMA		M–G		Lorenz	
Scaling	k	τ	k	τ	k	τ	k	τ
init	6	1	6	1	6	1	6	1
$\rho = 0.6$	4	1	4	1	2	1	2	3
$\rho = 0.95$	6	1	6	1	2	1	2	2
OG	6	1	6	1	2	1	2	2
ON	6	1	6	1	2	1	2	3

3.4 Analysis of information transfer in the reservoir

In the final step we wanted to have a closer look at the behaviour of the reservoir in different settings (introduced in section 3.1.1) regarding the information transfer. We generated one instance of the input weight vector $\mathbf{w}^{\mathbf{in}}$ and one instances of the recurrent weight matrix $\mathbf{W}$ (the way defined in section 3.1.1), and used them in all subsequent scalings and tasks. In the computations of the information measures in every task and scaling, we used the most frequent observations of the pair (k, τ) listed in table 3 as parameters of the target variable in the TE estimator and as global parameters in the AIS estimator. The source variable parameters in the TE estimator were set to $(1, 1)$, as we wanted to observe only how the most recent past of the source unit influences the target unit.

In order to quantify not only the global changes of information transfer due to scalings but also the changes in distribution of TE (when we consider the values of TE as a random variable), we computed relative entropy (Kullbeck–Liebler divergence) of TE, because the relative entropy is according to section 2.3.2 a more suitable measure of disorder (uncertainty) in a system when considering a continuous random variable then differential entropy. For the probability distribution $m(x)$ in equation 39, we choose the uniform distribution as it maximizes the differential entropy when no prior knowledge about distribution is available (It can be proved that the uniform distribution has the maximum differential entropy among all continuous distributions supported in the closed interval $[a, b]$. The proof follows from the non-negativeness property of Kullbeck-Liebler divergence). We choose the support of the uniform distribution to be the closed interval $[0; \max \mathrm{TE}]$ where $\max \mathrm{TE}$ is the maximum observed value of TE in a specific

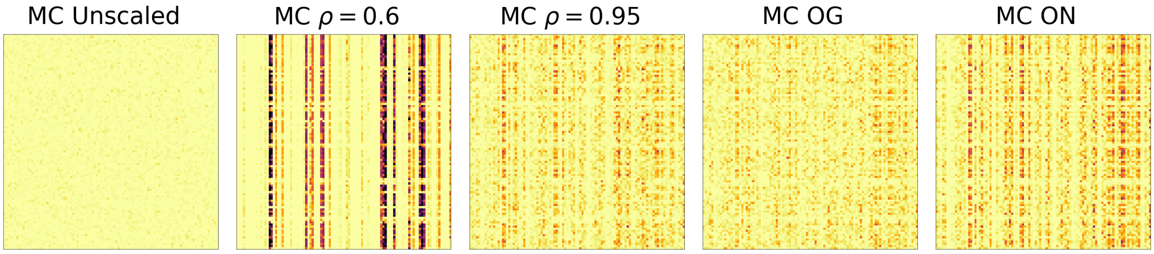

Figure 13: The matrices ($N \times N$) of TE values in the reservoir for different scalings in MC task. Rows of the matrix denote source units and columns denote target units. Dark color represents high values, light color represents low values.

task and scaling. The estimator of relative entropy takes the following form:

$$\left|\overline{H}(\mathrm{TE}_{\mathrm{RES}}^{(k,l)})\right| = \widehat{D_{\mathrm{KL}}}(\mathrm{TE}_{\mathrm{RES}}^{(k,l)} \,||\, \mathcal{U}(0; \max \mathrm{TE}_{\mathrm{RES}}^{(k,l)})) = \ln(\max \mathrm{TE}_{\mathrm{RES}}^{(k,l)}) - \widehat{H_{\mathrm{KL}}}(\mathrm{TE}_{\mathrm{RES}}^{(k,l)}) \quad (76)$$

where $\ln(\max \mathrm{TE}_{\mathrm{RES}}^{(k,l)})$ is the exact differential entropy of the uniform distribution with the support in $[0; \max \mathrm{TE}_{\mathrm{RES}}^{(k,l)}]$, $\widehat{H_{\mathrm{KL}}}(\cdot)$ is the Kozachenko-Leonenko entropy estimator(introduced in section 2.3.5) and $\mathrm{TE}_{\mathrm{RES}}^{(l,k)}$ is the distribution of the estimates of transfer entropies in the reservoir.

3.4.1 MC task

The visualisation of TE in reservoir for 5 different scalings in case of MC task is presented in figure 13. There is a noticeable change in the amount of TE by the transition from unscaled to scaled reservoir. Unscaled represents unstable regime, and scalings to spectral radius $\rho = 0.6$ and $\rho = 0.95$ represent stable regime and close to criticality respectively. Corresponding Lyapunov exponents λ values are listed in table 4. In case of $\rho = 0.6$ there are a few target neurons that have high values of TE (dark columns).The transition to criticality leads to the performance rising and global value of TE decreasing, as was observed in section 3.2.1. Concerning the OG and ON scalings there is an improved performance in case of OG and similar decrease of TE.

Figure 14 plots histograms of distribution of TE in corresponding reservoir scaling. We can observer the most visible changes from unscaled, to $\rho = 0.6$ and to $\rho = 0.95$. The quantification of the level of disorder of TE within the maximum value in every scaling is represented in table 4 by the relative entropy of TE. The unscaled case has the lowest value of relative entropy which means its distribution is closest to the distribution of the uniform distribution. We can interpret this result as the unscaled case has the highes level of disorder of TE among all scalings. With the transition to $\rho = 0.6$ the level of disorder decreases and further decreases with the close to criticality scaling. An interesting observation is that the OG scaling has the highest performance lowest TE and highest relative entropy of TE among the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON.

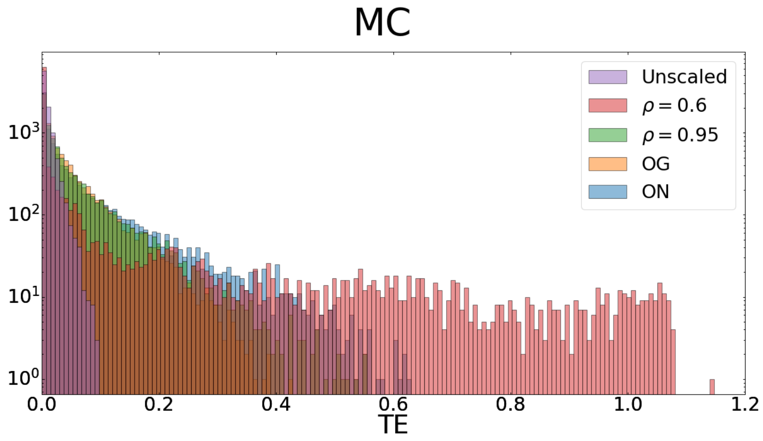

Figure 14: Histograms of TE distribution changes due to reservoir scaling in MC task (log scale is used).

Figure 15 provides the visualisation of the changes in AIS due to scaling for every unit. The results are consistent with results in section 3.2.1 in that the closer we are to critical point the higher the AIS value. An interesting observation is that the OG scaling has a minimal AIS value as well as TE value among the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON.

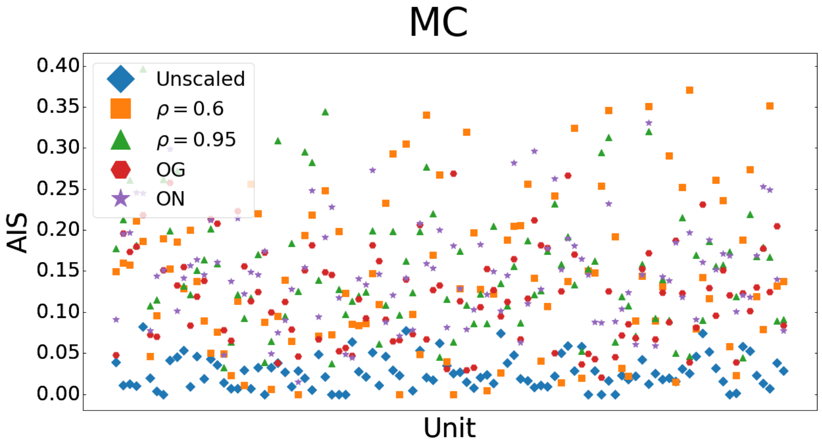

Figure 15: AIS values for every unit (N) in various reservoir scalings in MC task.

We also looked at distributions of reservoir activation signals (figure 16) for different scalings. The values of all the scalings settings except for unscaled case are within the

Table 4: Quantitative measures for MC task, in case of initialized, scaled and orthogonalized reservoirs.

MC task	Unscaled	$\rho = 0.6$	$\rho = 0.95$	OG	ON
MC	0.06	17.8	32.8	47.4	33.3
Average TE	0.009	0.092	0.048	0.042	0.062
Average AIS	0.027	0.146	0.151	0.12	0.148
Rel. entr. of TE	1.010	1.323	1.354	1.191	1.267
LE	0.53	-0.52	-0.06	-0.09	-0.16

range where the activation function can be approximated by a linear function what means that the reservoir does not benefit from the nonlinearities of tanh activation function.

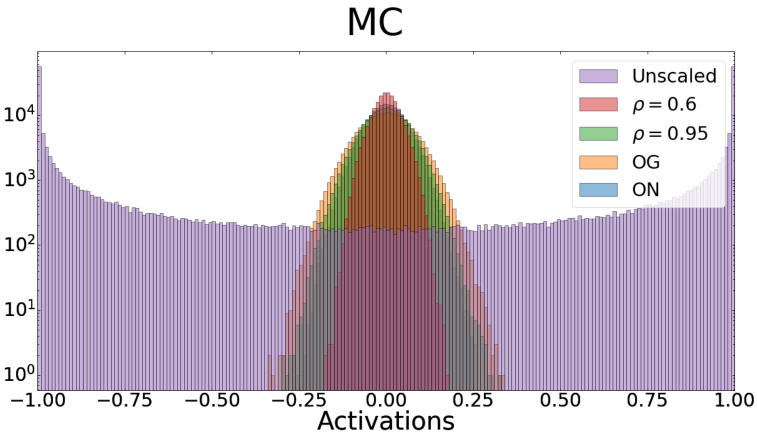

Figure 16: Reservoir neuron activation distributions for various reservoir scalings in MC task (log scale is used).

3.4.2 NARMA task

The results in NARMA task are very similar to the results in MC task. There is a visible change in the structure of TE matrices (Figure 17) with the transition from the unscaled to scaled cases. Also we can observe similar high TE target units(dark columns) for $\rho = 0.6$ and for $\rho = 0.95$, OG and ON cases and more spread distribution of TE in the reservoir. The Lyapunov exponent λ in all scalings is almost identical to the MC task.

The probability distribution of TE in the reservoir presented in Figure 17 look similar to the MC task as well. The biggest differences are between unscaled, scaled

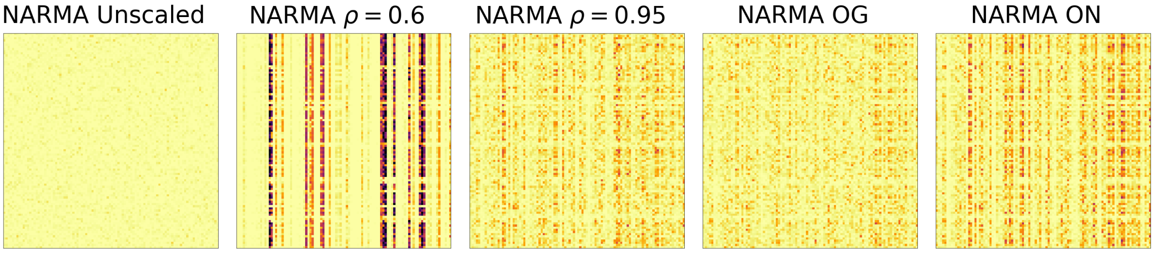

Figure 17: The matrices ($N \times N$) of TE values in the reservoir for different scalings in NARMA task. Rows of the matrix denote source units and columns denote target units. Dark color represents high values, light color represents low values.

to $\rho = 0.95$, OG and ON, and $\rho = 0.6$ scalings. These differences are supported by the statistical testing we have performed on the mean square errors (MSE) in every scaling using Kruskal–Wallis test. The differences in MSEs in the scalings $\rho = 0.95$, OG and ON are not statistically significant. Nevertheless from table 5 we get the same observation as in the MC task. In case of OG scaling the the performance is maximal, TE is minimal and relative entropy of TE is maximal among the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON.

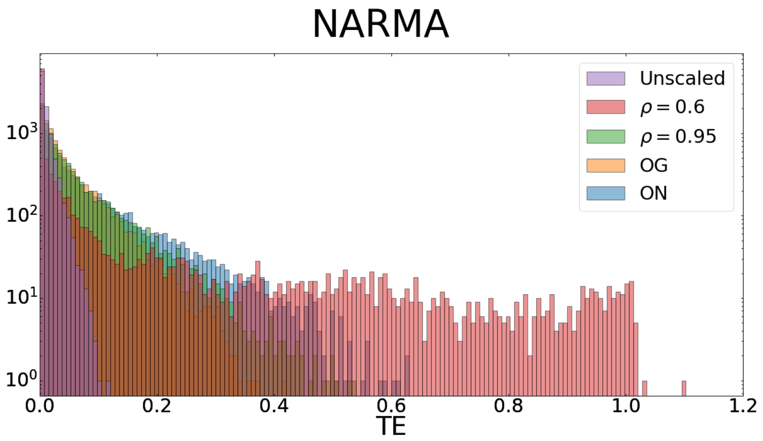

Figure 18: Histograms of TE distribution changes due to reservoir scaling in NARMA task (log scale is used).

The values of AIS for every unit in reservoir for every tested scaling shown in Figure 19, shows again qualitative similarities of MC and NARMA tasks. There are visible changes when comparing unscaled and scaled cases. In this case the lowest mean value of AIS among the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON is observed in case of $\rho = 0.95$.

Concerning the distribution of reservoir unit activations (Figure 20), it is even more

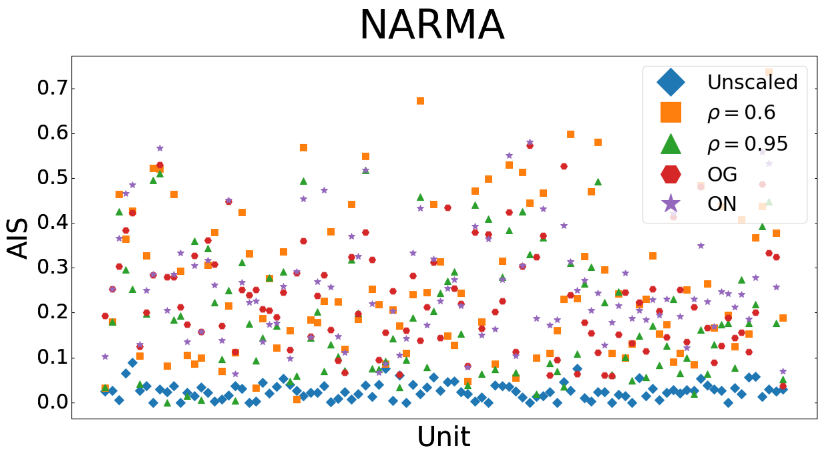

Figure 19: AIS values for every unit (N) in various reservoir scalings in NARMA task.

Table 5: Quantitative measures for NARMA task, in case of initialized, scaled and orthogonalized reservoirs.

NARMA	Unscaled	$\rho = 0.6$	$\rho = 0.95$	OG	ON
NRMSE	2.07	0.98	0.83	0.82	0.84
Average TE	0.0093	0.088	0.053	0.044	0.067
Average AIS	0.025	0.269	0.191	0.232	0.257
Rel. entr. of TE	1.175	1.642	1.271	1.183	1.194
LE	0.52	-0.53	-0.062	-0.089	-0.16

focused when compared to the MC task. This means that the reservoir does not benefit from the nonlinearities of the tanh activation function for the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON, and could be replaced by a linear function.

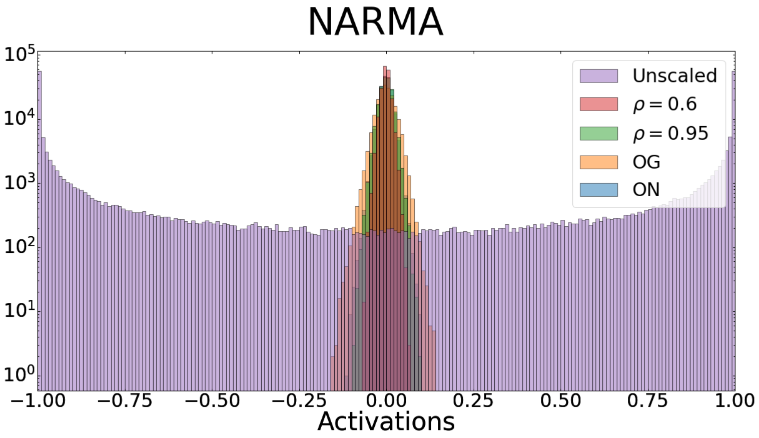

Figure 20: Reservoir neuron activation distributions for various reservoir scalings in NARMA task (log scale is used).

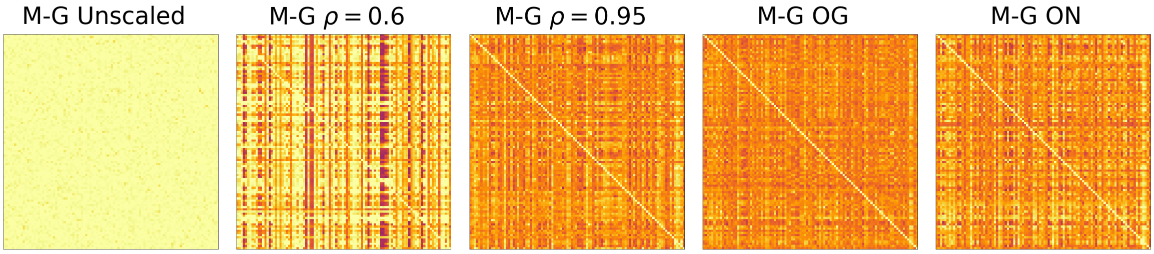

Figure 21: The matrices ($N \times N$) of TE values in the reservoir for different scalings in M–G task. Rows of the matrix denote source units and columns denote target units.

3.4.3 M–G task

In the case of M–G task there are noticeable differences in quality and quantity with the transition from scaled to unscaled when compared to the previous tasks. This supports our findings from sections 3.2 and 3.3. There is only negligible global value of TE in the unscaled reservoir, but the minimal value among scaled reservoirs is for $\rho = 0.6$ as opposed to OG in MC task and $\rho = 0.95$ in NARMA task. The same goes for performance. We observe the best performance for the scaling $\rho = 0.6$, which is in the stable region of the Lyapunov exponent spectrum. We have performed a Kruskal–Walis test for differences in MSE among 5 scalings and found statistically significant diferences in unscaled, $\rho = 0.6$ and $\rho = 0.95$. For the scalings $\rho = 0.95$, OG, ON there is no significant difference.

The distribution of TE in various scalings (Figure 22) shows some interesting similarities and differences when compared to MC and NARMA tasks. The maximum values of TE are lower than in MC and NARMA tasks (max TE ≈ 0.6 vs. max TE ≈ 1.1 respectively) but the distributions of unscaled and $\rho = 0.6$ look visually similar to MC and NARMA cases. Also quantitatively, the relative entropy of TE (table 6) how smaller values compared to MC and NARMA tasks, meaning that the distribution of TE is more closer to uniform distribution. This can be observed visually in the TE matrices (Figure 21). Similarly to previous tasks the scaling with the best performance ($\rho = 0.6$) has the lowest value od mean TE and lowest value of relative entropy among the scalings $\rho = 0.6$, $\rho = 0.95$, OG, ON.

The values of AIS for individual units in various scaling are presented in Figure 23. Similarly to previous tasks, there is a noticeable shift from unscaled to scaled reservoirs but in M–G task the values of AIS are much higher (max AIS ≈ 3.4) than in MC and NARMA tasks (max AIS ≈ 0.4 and max AIS ≈ 0.7). The best performing scaling

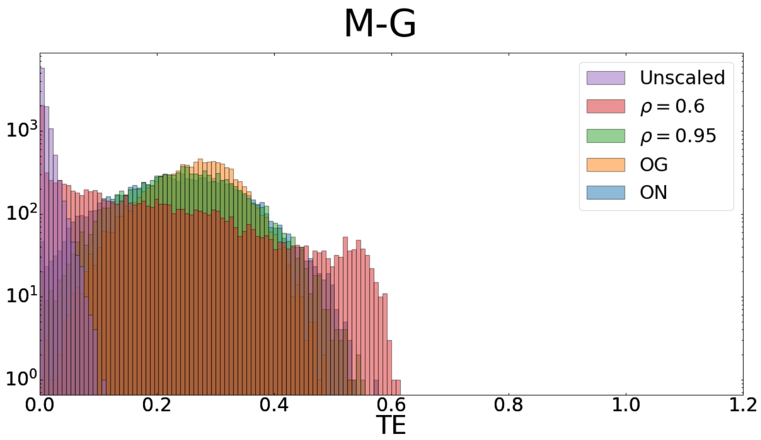

Figure 22: Histograms of TE distribution changes due to reservoir scalings in M–G task (log scale is used).

($\rho = 0.6$) has the highest mean value of AIS among the scalings $\rho = 0.6$,$\rho = 0.95$, OG and ON. This observation extends our findings from section 3.2. Compared to the MC and NARMA tasks, the best performing scaling had the lowest AIS among scaled instances.

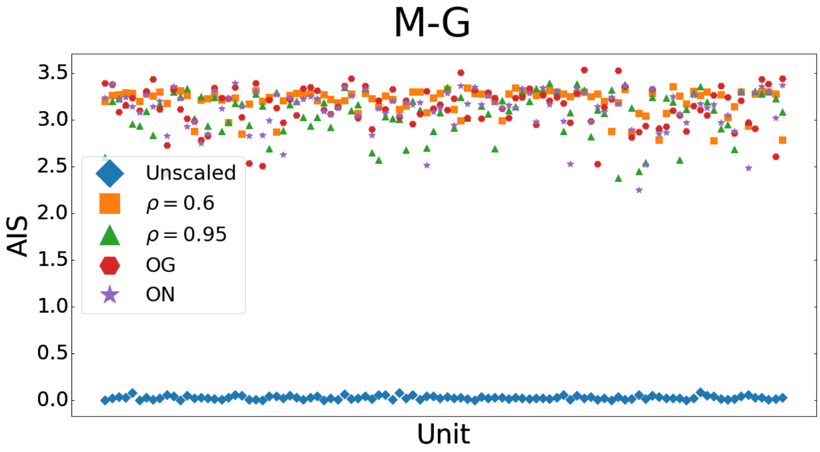

Figure 23: AIS values for every unit (N) in various reservoir scalings in M–G task.

The reservoir activations are similarly distributed as in previous cases (Figure 24). That is the activations in scaled reservoirs are very focused around zero meaning the operate in linear mode and do not benefit from the nonlinearities of tanh activation function.

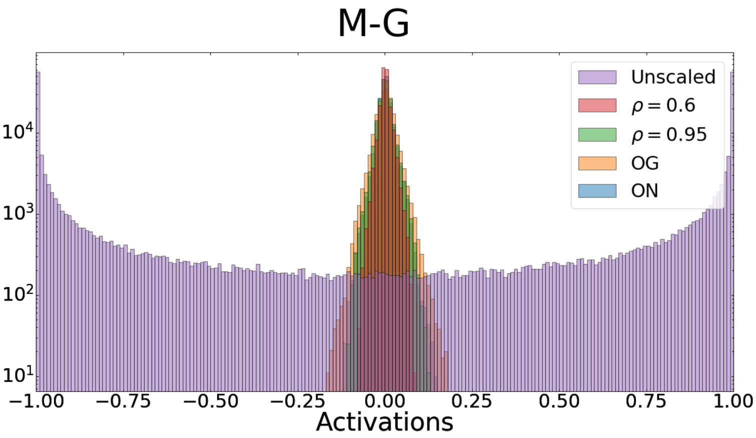

Figure 24: Reservoir neuron activation distributions for various reservoir scalings in M–G task (log scale is used).

Table 6: Quantitative measures for M–G task, in case of initialized, scaled and orthogonalized reservoirs.

Mackey–Glass	Unscaled	$\rho = 0.6$	$\rho = 0.95$	OG	ON
NRMSE	1.12	0.00025	0.00026	0.0003	0.00026
Average TE	0.0093	0.15	0.25	0.26	0.24
Average AIS	0.025	3.204	3.073	3.142	3.115
Rel. entr. of TE	1.166	0.315	0.391	0.588	0.328
LE	0.53	-0.52	-0.062	-0.09	-0.16

3.4.4 Lorenz task

The results in the Lorenz task confirm our findings in section 3.3 that there are two complexity classes in tested tasks. MC and NARMA, and M–G and Lorenz. The result regarding quantitative and qualitative properties of TE in Lorenz task are very much like the results in M–G task. The structure of TE matrices for various scalings (Figure 25) resembles the TE matrices in M–G task, just with overall lower TE values. The best performance is observed in the $\rho = 0.6$ scaling just as in M–G task, which means better performance in stable regimes than closer to criticality. The Kruskall-Walis test showed statistically significant differences between all scalings.

The distributions of various scalings in Lorenz task presented in Figure 25 show generally lower values of TE and also mean TE when compared to M–G task. Also the structure of the distribution has some dissimilarities for scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON. Regarding the relative entropy of TE shown in Table 7, in case of the scaling

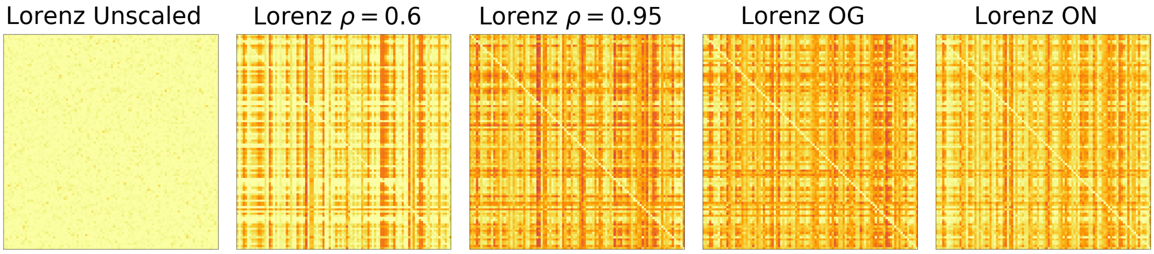

Figure 25: The matrices ($N \times N$) of TE values in the reservoir for different scalings in Lorenz task. Rows of the matrix denote source units and columns denote target units.

which has the largest performance ($\rho = 0.6$) has also the largest relative entropy of TE among the scalings $\rho = 0.6$, $\rho = 0.95$, OG and ON. This observation is a difference to NARMA and M–G prediction tasks where the scaling with the biggest performance has the smallest relative entropy of TE.

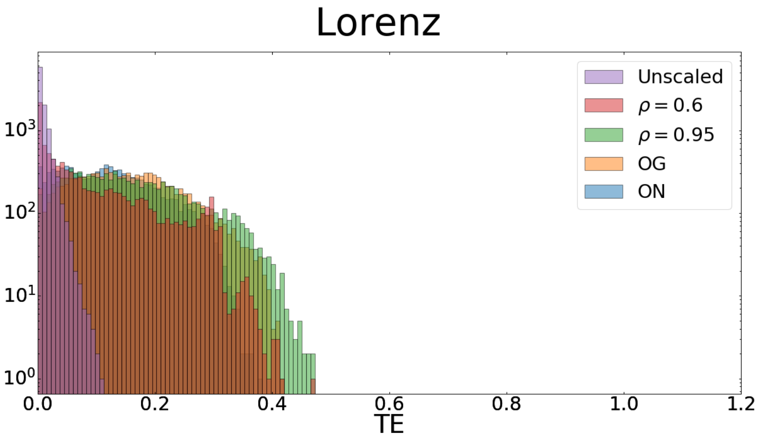

Figure 26: Histograms of TE distribution changes due to reservoir scaling in Lorenz task (log scale is used).

AIS global and individual unit values, shown in Figure 27, are again higher when compared to MC and NARMA tasks. Overall the plot is very similar to the case of M–G task and also the scaling with the best performance ($\rho = 0.6$) has the largest mean value of AIS among alls scalings.

The values of reservoir unit activations observed in Figure 28 for all scalings cover the whole range of the tanh activation fucntion. This is a change when comparing to previous tasks. Still the most frequent activation values are around zero. This could be explained by the range of the driving signal and that we used the same input matrix $\mathbf{w}^{\mathbf{in}}$ which is not an optimal input matrix for the Lorenz task but a compromise for comparability reasons.

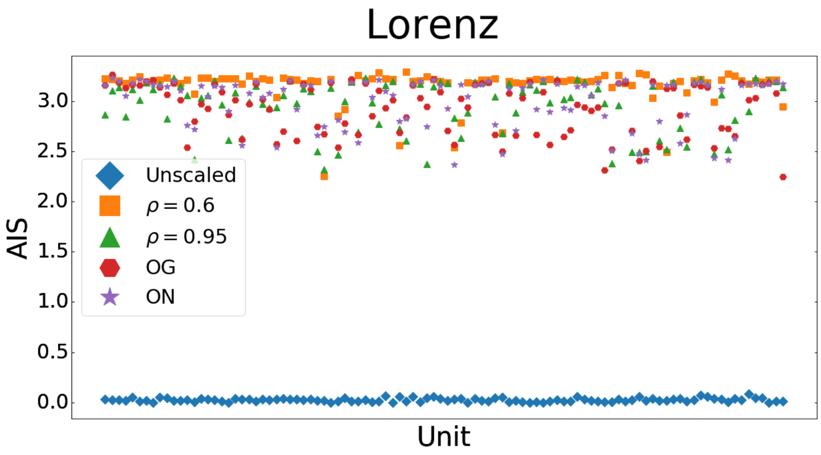

Figure 27: AIS values for every unit (N) in various reservoir scalings in Lorenz task.

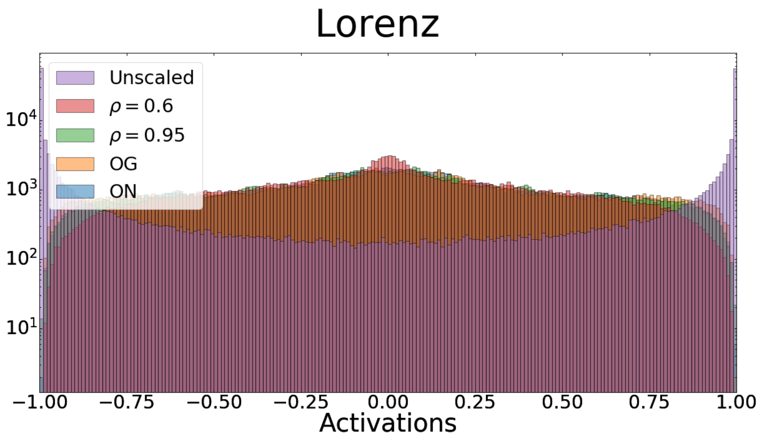

Figure 28: Reservoir neuron activation distributions for various reservoir scalings in Lorenz task (log scale is used).

Table 7: Quantitative measures for Lorenz task, in case of initialized, scaled and orthogonalized reservoirs.

Lorenz	Unscaled	$\rho = 0.6$	$\rho = 0.95$	OG	ON
NRMSE	0.56	0.00012	0.0013	0.0034	0.00097
Average TE	0.009	0.087	0.16	0.15	0.12
Average AIS	0.025	3.157	2.971	2.932	3.002
Rel. entr. of TE	1.148	0.694	0.254	0.221	0.262
LE	0.52	−0.74	−0.28	−0.35	−0.32

3.4.5 Conclusion

The results presented in this section confirm results in sections 3.2 and 3.3, that there are two complexity classes in relation to driving signals. This should come as no surprise since the driving signals in MC and NARMA tasks are stochastic, and in M–G and Lorenz are deterministic in nature. Performance wise M–G and Lorenz tasks do not benefit from criticality, as is the case of MC and NARMA tasks, but operate better in stable regimes.

When comparing among task classes with complexity in mind, there is a general rise in performance, TE and AIS, and decline in relative entropy of TE in reservoir between NARMA and M–G/Lorenz tasks. On the other hand when comparing scalings within each task, the relationship is not so clear. In the stochastic complexity class (MC/NARMA) we observe rise of performance, and decline of TE and AIS. In the deterministic complexity class (M–G/Lorenz), rise in performance and AIS, decline in TE. These results support the findings in section 3.2 regarding the behavior of information measures in relation to stability in stable region. Additional information with regard to distribution of TE in the reservoir, shows that lower values of relative entropy of TE are accompanied with better performance when comparing scalings in MC, NARMA and M–G tasks. This is not the case in Lorenz task.

These results point to the hypothesis that there might be a relationship between performance, global TE and distribution of TE in the reservoir. More precisely that not only the amount of information transfer between units is important but also that it should be more uniformly distributed in the reservoir. These claims should be throughoutly investigated.

4 Discussion

References

[1] J. Boedecker and O. Obst and J. Lizier and N. Mayer and M. Asada, "Information processing in echo state networks at the edge of chaos", Theory in Biosciences, vol. 131, pp. 205–213, 2012.

[2] S. Haykin, Neural Networks and Learning Machines (3rd Edition), Prentice Hall, 2009.

[3] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks", German National Research Institute for Computer Science, Tech. Rep. GMD Report 148, 2001.

[4] H. Jaeger, "Short term memory in echo state networks", German National Research Institute for Computer Science, Tech. Rep. GMD Report 152, 2001.

[5] K. Hornik, "Approximation capabilities of multilayer feedforward networks", Neural networks, vol. 4, pp. 251-257, 1991.

[6] M. Lukoševičius, "A practical guide to applying echo state networks", In: G. Montavon, G. B. Orr, and K.-R. Müller (eds.) Neural Networks: Tricks of the Trade, 2nd ed. Springer,pp. 659-686, 2012.

[7] M. Cencini, F. Cecconi, A. Vulpiani, Chaos: From Simple models to complex systems (Series on Advances in Statistical Mechanics) (Volume 17), World Scientific Publishing Co, 2009.

[8] I. Farkaš and R. Bosák amd P. Gergeľ,"Computational analysis of memory capacity in echo state networks", Neural Networks, vol. 83, pp. 109–120, 2016.

[9] D. MacKay, Information Theory, Inference, and Learning Algorithms (fourth printing), Cambridge University Press, 2005.

[10] C. E. Shannon, "A mathematical theory of communication", Bell Systems Technical Journal, vol. 27, pp. 379–423, 1948.

[11] M. Paluš, V. Komárek, Z. Hrnčíř, K. Šteřbová, "Synchronization as adjustment of information rates: Detection from bivariate time series", Physical Review E, vol. 63, p. 046211, 2001.

[12] T. Schreiber, "Measuring information transfer", Physical Review Letters, vol. 85, no. 2, pp. 461–464, 2000.

[13] T.ăBossomaier, L.ăBarnett, J. T.ăLizier, An Introduction to Transfer Entropy, Springer, 2016.

[14] M. Wibral, R. Vicente, J. T.Lizier, Eds., Directed Information Measures in Neuroscience. Springer, 2014.

[15] A. Kraskov, H. Stögbauer, P. Grassberger, "Estimating mutual information,: Physics Reviews E, vol. 69, p. 066138, 2004.

[16] G. Gómez-Herrero, W. Wu, K. Rutanen, M. C. Soriano, G. Pipa, R. Vicente, "Assessing coupling dynamics from an ensemble of time series", Entropy, vol. 17, no.4, pp. 1958–1970.

[17] M. Ragwitz, H. Kantz, "Markov models from data by simple nonlinear time series predictors in delay embedding spaces",Physics Reviews E, vol. 65, no. 5, p. 056201, 2002.

[18] J. Geweke, "Measurement of linear dependence and feedback between multiple time series", Journal of American Statistical Association, vol. 77, no. 378, pp. 304–313, 1982.

[19] L. Barnett, A. B. Barrett, A. K. Seth, "Granger causality and transfer entropy are equivalent for Gaussian variables", Physical Review Letters, vol. 103, no. 23, p. 238701, 2009.

[20] E. T. Jaynes, Probability Theory: The Logic of Science, Cambridge University Press, 2003.